



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1965

A computer simulation of an aircraft penetration over hostile terrain.

Harrison, James L.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/24638>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NPS ARCHIVE
1965
HARRISON, J.

A COMPUTER SIMULATION OF AN AIRCRAFT
PENETRATION OVER HOSTILE TERRAIN

JAMES L. HARRISON

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

A COMPUTER SIMULATION OF AN AIRCRAFT
PENETRATION OVER HOSTILE TERRAIN

* * * * *

James L. Harrison

A COMPUTER SIMULATION OF AN AIRCRAFT
PENETRATION OVER HOSTILE TERRAIN

by

James L. Harrison

Captain, United States Marine Corps Reserve

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
OPERATIONS RESEARCH

United States Naval Postgraduate School
Monterey, California

1 9 6 5

NPS Archive

1965

Harrison, J.

~~1965~~

Library
U. S. Naval Postgraduate School
Monterey, California

A COMPUTER SIMULATION OF AN AIRCRAFT
PENETRATION OVER HOSTILE TERRAIN

by

James L. Harrison

This work is accepted as fulfilling
the thesis requirements for the degree of

MASTER OF SCIENCE

IN

OPERATIONS RESEARCH

from the

United States Naval Postgraduate School

ABSTRACT

A computer simulation of an aircraft making a low level penetration over hostile terrain is described. The model discussed is intended as a tool for the determination of appropriate defensive tactics for aircraft upon receiving fire from conventional ground weapons. In any computer model of a ground to air combat environment it is essential to have a realistic simulation of the terrain. The method used in this simulation is discussed and shown to be extremely accurate.

TABLE OF CONTENTS

Section	Title	Page
1.	Introduction	1
2.	Design of the Model	1
3.	Aircraft Movement	3
4.	Determination of the Aircraft's Survival Probability	4
5.	Terrain Simulation	6
6.	Analysis of the Model	13
7.	Conclusions and Recommendations	26
8.	Bibliography	27
Appendix		
I.	Flow Charts of PENAIR	28
II.	FORTTRAN listing of PENAIR	60
III.	Inputs and outputs of PENAIR	76
IV.	FORTTRAN listing of TERRAIN	86
V.	Inputs and outputs of TERRAIN	121



LIST OF ILLUSTRATIONS

Figure	Page
1. The Arrangement of the Individual Squares of Terrain Simulated in the PENAIR Program	3
2. Determination of the Aircraft's Time in View of a Weapon	5
3. Actual Contour Map of Grid Square 7174, U. S. Army Hunter-Liggett Military Reservation	11
4. Computer Simulation of Grid Square 7174, U. S. Army Hunter-Liggett Military Reservation	11
5. Representative Fit at the Junction of Two Grid Squares	12
6. A Straight and Level Flightpath	15
7. A Flightpath of an Aircraft Climbing Straight Ahead	16
8. A Flightpath of an Aircraft Turning Away from a Firing Weapon	17
9. A Nap of the Earth Flightpath	18
10. Comparison of Rates of Climbs and Descents in a Nap of the Earth Flight at a Constant Speed of 100 Kts.	21
11. Comparison of Aircraft Speeds on a Nap of the Earth Flight	22
12. Comparison of the Aircraft's Radius of Turn at 3 Airspeeds, 70, 90, and 110 Kts.	23
13. Computer Simulation of an Actual Flightpath Flown by a UH-1B Helicopter	24
14. First Page of Output from the PENAIR Program	84
15. Second Page of Output from the PENAIR Program	85
16. Distortion Caused by Projection of a Map Grid Square	127
17. A Sample Data Deck for the TERRAIN Program	128
18. A Sample of a Plotted Map from the TERRAIN Program	130

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

THE JOURNAL OF THE

1. Introduction

PENAIR is a time step computer simulation of an aircraft flight over hostile terrain. It is an analytic computer war game in which one aircraft and a maximum of two hundred weapons are the principles. Any piece of actual terrain may be simulated using a contour map of the area to obtain the input parameters.

The simulation is intended as a tool for the determination of appropriate defensive tactics for an aircraft upon receiving fire from conventional ground weapons. Using the characteristics of the aircraft and of the weapons the model determines the survival probability of the aircraft on any of several flightpaths. There are no restrictions on the aircraft's speed, altitude or other characteristics and therefore this model is equally suitable for helicopters or jets.

The aircraft flightpath can be predetermined or it can be generated by the model. Upon receiving fire the aircraft can be programmed to execute any of 18 evasive maneuvers. A comparison of the survival probabilities can then be made thus giving an indication of the appropriate defensive maneuver to employ.

The terrain simulation and generated flightpaths appear to be extremely realistic. The terrain is simulated by fitting, in the least squares sense, a polynomial function of two independent variables (x, y) to values of a dependent variable specified at points on a rectangular grid in the plane of the independent variables. This is accomplished by a separate computer program, TERRAIN, which generates the necessary inputs for PENAIR.

2. Design of the Model

PENAIR is a computer simulation of a single aircraft making a low level penetration over terrain containing hostile small arms and conventional anti-aircraft weapons. Any actual or fictitious terrain may be

simulated. No offensive aircraft weapons are considered although 18 different defensive maneuvers such as increasing speed and diving for the deck may be performed. Ten types of ground weapons may be used with a maximum of 20 weapons per type. The program computes the aircraft's time in view of each weapon, the probability of the aircraft surviving each weapon and the aircraft's total survival probability. The simulation runs in discrete time steps using any time interval desired. The program does not utilize Monte Carlo methods; aircraft survival probability for a specified flightpath over a specified terrain is computed in a single run. Flow chart 1.0 of Appendix I, page 30, indicates the general logic of the PENAIR program.

PENAIR is written in FORTRAN-63 for the CDC 1604 computer. The PENAIR program is included as Appendix II. A separate program, TERRAIN, is utilized to supply inputs to PENAIR for terrain simulation. The terrain input program supplies the necessary data on punched cards for the PENAIR program. The TERRAIN program is also programmed in FORTRAN-63 for the CDC 1604 and is included as Appendix IV.

The terrain used in the PENAIR program may be divided into as many as 12 separate squares for simulation. Each of the 12 individual squares is approximated by a separate polynomial. The 12 squares may be of any common size. Figure 1 illustrates the arrangement of these squares. The coordinate system has been established such that the lower left corner is the origin with increasing x to the right and increasing y upward. It is possible to use actual military grid coordinates since the origin may take any desired x and y values.

A special PENAIR output is a set of punched cards to be used as input to a graph plot routine for an off-line plotter. This routine then draws the actual flightpath over the simulated terrain. Output from this routine is included in Section 6.

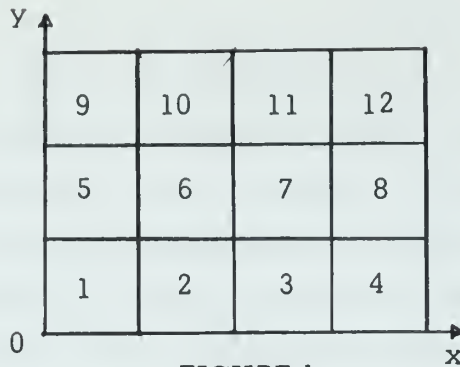


FIGURE 1

THE ARRANGEMENT OF THE INDIVIDUAL
SQUARES OF TERRAIN SIMULATED IN
THE PENAIR PROGRAM

3. Aircraft Movement

The PENAIR model has provisions for three types of simulated aircraft flightpaths. The first of these flightpaths allow the PENAIR program to select one of 18 different evasive maneuvers for the aircraft upon receiving fire. The 18 evasive maneuvers consist of six basic maneuvers with three possible speed changes for each. The six basic maneuvers are: (1) continue flying straight ahead at the same altitude, (2) commence a climb straight ahead at the aircraft's maximum rate of ascent, (3) turn away from the weapon remaining at the same altitude, (4) turn away and commence a maximum rate climb, (5) dive for the deck and commence nap of the earth (contour) flight straight ahead, or (6) turn away from the weapon, dive for the deck and commence nap of the earth flight. For each of the above maneuvers the aircraft may decrease speed, remain at the same speed or increase speed. All of the above maneuvers are generated using the input characteristics of the aircraft being simulated. It should be realized by the PENAIR user however that only one basic maneuver



[The following text is extremely faint and illegible. It appears to be a list or a series of entries, possibly a table of contents or a list of items. The text is too blurry to transcribe accurately.]

may be employed in one computer run. The maneuver employed is determined by the numerical value of the input flag IMARK.

The other two simulated aircraft flightpaths are preplanned. In the first of these, the flightpath is completely determined in advance and read into the PENAIR program from data cards. In the second, the x and y coordinates of points along the flightpath are read in to the PENAIR program. From these points the PENAIR program computes a map of the earth flightpath for the aircraft over the simulated terrain. In both of these cases, the PENAIR program makes no deviation from the pre-planned flightpath and the aircraft makes no use of the evasive maneuvers. A complete description of the inputs needed to regulate aircraft movement is included in Appendix III.

4. Determination of the Aircraft's Survival Probability.

The major output of the PENAIR program is the probability of aircraft survival. This survival probability is computed as a function of weapon type, range and time in view. The necessary inputs are weapon characteristics and the single time-interval kill probabilities for each weapon. Given these kill probabilities, the probability of an aircraft surviving any single weapon is computed as:

$$P (\text{Survival}) = (1 - P_1) (1 - P_2) \dots (1 - P_n)$$

where P_i is the probability of kill for one time interval at range i for a given weapon. The total probability of aircraft survival is then computed as the product of the individual probabilities of survival for each weapon. (See Flow Chart 5.1, Appendix I, page 48). It is assumed that the probability of a kill by one weapon is independent of all other weapons. The number of time intervals the aircraft receives fire depends on the time in view, minimum and maximum weapon range, and the weapon acquisition, firing, and reloading times.

Time in view is the one parameter to be determined. The method used by this simulation to determine time in view will be explained using Figure 2 and Flow Chart 7.0 of Appendix I, page 49.

The aircraft is considered in view of a weapon if every point on the line of sight WA between the weapon position W, and the aircraft A, is higher than the corresponding point of the terrain. To determine this the x,y coordinates of the point on the line TR, 10 meters from the weapon are used to calculate the elevation of the terrain at that point. This elevation is then compared with the height of the line of sight at those coordinates. If the terrain is lower, another point 50 meters farther along the line TR is chosen and the same comparison is made. If the terrain elevation is below the line of sight elevation at this point this procedure is repeated every 50 meters. This process is continued until it is determined that the aircraft is in view of the weapon or that the terrain elevation exceeds the line of sight. If the latter occurs the aircraft is considered not in view of this weapon at this time. This process is repeated for all weapons and then the aircraft is advanced one time interval. The number of consecutive time intervals the aircraft remains in view of any one weapon is then used to compute the current total amount of time in view per weapon.

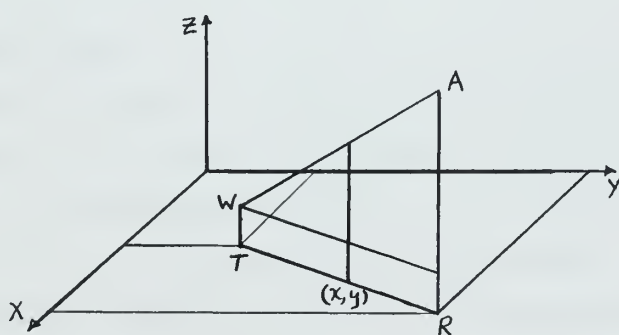


FIGURE 2
DETERMINATION OF THE AIRCRAFT'S TIME
IN VIEW OF A WEAPON

The first part of the paper discusses the importance of the study and the objectives of the research. It highlights the need for a comprehensive understanding of the subject matter and the role of the researcher in this process. The second part of the paper presents the methodology used in the study, including the selection of the sample and the data collection techniques. The third part of the paper discusses the results of the study and the conclusions drawn from the data. The fourth part of the paper discusses the implications of the study and the recommendations for future research.

The study was conducted in a systematic and rigorous manner, following the principles of scientific research. The data collected was analyzed using statistical methods, and the results were presented in a clear and concise manner. The study has contributed to the understanding of the subject matter and has provided valuable insights into the field. The findings of the study have implications for practice and for future research.

The study was conducted in a systematic and rigorous manner, following the principles of scientific research. The data collected was analyzed using statistical methods, and the results were presented in a clear and concise manner. The study has contributed to the understanding of the subject matter and has provided valuable insights into the field. The findings of the study have implications for practice and for future research.



Two other methods of determining if the aircraft was in view of the weapon were tested. Elevation comparisons between line of sight and terrain were made at 25 meter increments on the line of sight WA, and at 25 meter increments along TR. No significant differences were noted in the times of view computed by the 3 methods mentioned. However, the method outlined in the previous paragraph greatly reduces computer running time for the PENAIR program.

After the aircraft has been in view of a weapon for the required acquisition time, the weapon minimum and maximum ranges are checked. If the aircraft is within firing range, the weapon commences firing and the probability of survival is determined for this weapon in this time interval. The probabilities of aircraft survival for each weapon are then computed for all following time intervals until the aircraft either goes out of view and/or range or until the weapon has expended its ammunition. In both cases no further probabilities are computed until the weapon is reloaded or until the aircraft comes back in view and within range. It is assumed that after reloading reacquisition requires one-half the acquisition time. If the aircraft goes out of view the weapon must wait its full acquisition time after the aircraft is again in view.

5. Terrain Simulation.

In any computer model of a ground combat environment it is essential to have a realistic simulation of the terrain in order to determine the mask angles of ground weapon positions. To do this the simulation must provide the elevation of any point in the combat area. Present computer methods of terrain simulation either use very specialized equipment or involve prohibitive amounts of computer storage. The common method is to divide the terrain of interest into equal areas and assign an appropriate elevation to each. The assigned elevation might be the average elevation of the area. In this method it is assumed that all points in a given area, for the purpose of simulation, have the same elevation. This is a

crude approximation of terrain features for large areas. The approximation approaches reality as the size of the areas decreases. However, since the number of the areas considered is inversely proportional to the size of the areas, as the approximation of the terrain approaches reality the computer storage required becomes very large.

For the model developed here the terrain features are expressed in polynomial form, $z=f(x,y)$. The general problem is that of fitting points in three dimensions to a surface represented by the polynomial, $z=f(x,y)$. A method of forming this polynomial using the principle of least squares, given the set of data points $\{(x,y,z)\}$, had been programmed in FORTRAN-60 for the CDC 1604 [2]. This program was converted to FORTRAN-63 and a successful attempt was made in fitting a polynomial to a fairly rough surface (mountainous terrain.) The polynomials that best fit mountainous terrain whose elevations ranged from 800 to 1300 feet have approximately 400 terms with the maximum degree of both x and y near 20.

The program designed to fit the polynomial to the terrain required the input points be from an equally spaced grid in the x,y plane. For this thesis, areas of 1000 meters on a side were arbitrarily chosen from maps of actual terrain. In order that the fit be acceptable on the edges of the grid square, an area of at least 1200 x 1200 meters with the desired 1000 meter square centered inside had to be considered. Consecutive x and y coordinates of each individual contour line were recorded and a program was written to produce equally spaced elevations from the consecutive data points. However, because of the great amount of storage required, these two programs could only be incorporated into one by the use of overlays. An overlay is a feature of FORTRAN-63 designed to increase the use of the core storage of the computer by using magnetic tape as auxiliary storage.

The program to compute $z=f(x,y)$ from the consecutive (x,y) points of actual contour lines consists of a main program and three overlays.

The sole purpose of the main program is to call the individual overlays into operation. The first overlay, MAPPING, computes the equally spaced elevation points using the (x,y) points from the contour lines as input. The second overlay, SURFIT, by using the points computed in the first overlay, computes the polynomial $z = f(x,y)$ and records the coefficients on punched cards. The second overlay also computes the elevations of a grid for use in the third overlay. The third overlay, PLOTTER, generates data and stores it on a magnetic tape suitable for a CDC-160 program used to drive a CalComp 165 graph plotter. The contour map drawn can then easily be compared to the original map to estimate the goodness of fit of the polynomial. The drawing of this map is optional and is controlled by an input parameter to the main program. A detailed description of each overlay follows.

Overlay 1 determines the elevation of the terrain at equally spaced points of a grid in the x,y plane. The program was written by T. K. Rodeburg of the Norwegian Defense Institute for Stanford Research Institute, Fort Ord, California. This overlay uses as input the lists of the x and y coordinates at successive points on each contour line read from an enlarged map of the actual terrain. The following steps are taken by this overlay in the process of determining the elevation of the grid points:

- (1) The point coordinates are corrected for distortion caused by unequal magnification of the original map.

- (2) Equally spaced vertical lines are determined and the coordinates of the intersection of the contour lines and these vertical lines are calculated.

- (3) The elevation at these intersections are calculated by linear interpolation.

- (4) By further linear interpolation, elevations at equally spaced points on the vertical lines are found, thus obtaining the elevations of points on a grid in the x,y plane.

The first part of the paper discusses the importance of the
 research and the objectives of the study. It also outlines the
 methodology used in the study and the data sources. The second
 part of the paper presents the results of the study and discusses
 the implications of the findings. The third part of the paper
 concludes the study and provides recommendations for future research.

Table 1: Summary of the study results									
Variable	Mean	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis	Shapiro-Wilk	Normality	Significance
Variable 1	1.2	0.5	0.0	2.0	0.1	1.5	0.95	Yes	0.001
Variable 2	1.5	0.6	0.0	2.5	0.2	1.8	0.92	Yes	0.002
Variable 3	1.8	0.7	0.0	3.0	0.3	2.0	0.89	Yes	0.003
Variable 4	2.1	0.8	0.0	3.5	0.4	2.2	0.86	Yes	0.004
Variable 5	2.4	0.9	0.0	4.0	0.5	2.5	0.83	Yes	0.005

The results of the study show that the variables are normally distributed and that the differences between the groups are statistically significant. The findings suggest that the research objectives have been achieved and that the methodology used in the study is valid. The implications of the findings are discussed in the third part of the paper.

The study concludes that the research objectives have been achieved and that the methodology used in the study is valid. The findings suggest that the research objectives have been achieved and that the methodology used in the study is valid.

(5) Using the original data points, the above interpolation procedure is repeated at equally spaced horizontal lines.

(6) The average of the two values thus obtained is used as the elevation at each point of the grid.

Overlay 2 fits, in the least squares sense, a polynomial function of two independent variables, x and y , to values of a dependent variable, z , specified at points on a rectangular grid in the plane of the independent variables. That is, given a set of ground elevations, z at grid coordinates

x and y , this overlay finds the polynomial, $\sum_{n=0}^N \sum_{m=0}^M A_{nm} X^n Y^m$, which best fits the data. The use of orthogonal polynomials leads to a particularly simple system of linear equations rather than the ill-conditioned system which arises from the usual least squares normal equations. It also provides a measure of the improvements resulting from each new term included which further leads, in this algorithm, to an automatic selection of a "best" degree polynomial function as determined by Gauss' criterion [1]. It therefore determines the optimum values of N and M as well as the coefficients.

This algorithm was programmed in FORTRAN-60 for the CDC 1604 by Mrs. Jean Bow of the USNPGS from an ALGOL program [2]. Minor modifications were made to convert it to a FORTRAN-63 overlay. The FORTRAN-60 version is a very general program whereas the overlay has had many of the generalities removed for ease of handling inputs which in turn reduced the computer running time significantly. A FORTRAN listing of this program appears in Appendix IV.

The following steps are taken by Overlay 2 in the process of determining an approximating polynomial:

- (1) the variables are normalized by subtracting off their means,
- (2) the orthogonal polynomials are evaluated,
- (3) the contribution of each orthogonal polynomial to the minimization of the residuals is evaluated,

(4) Gauss' criterion is applied to determine the degree of polynomial which yields the closest fit to the given data.

(5) the orthogonal polynomial coefficients are evaluated,

(6) the dependent variables are evaluated using the approximating polynomial, and

(7) the terrain elevations utilizing a 50 meter grid are evaluated for input into Overlay 3.

Overlay 3 generates lists of x and y coordinates of successive points on each contour line and stores these coordinates on an output tape in a form suitable for a CDC-160 program used to drive a CalComp 165 plotter. This tape is used as an input tape for the plotter which then reconstructs the terrain by connecting the points with straight line segments.

The FORTRAN-63 program used in Overlay 3 was originally developed for use with the IBM 7090 system and an x,y plotter, to give a complete traditionally styled contour map. This program was converted to FORTRAN-63 by Mr. R. R. Hillery of the USNPGS, for use on the CDC 1604. Subroutines used to compute the points to be plotted were provided by Dr. M. O. Dayhoff of the National Biomedical Research Foundation, Silver Spring, Maryland [3]. Subroutines used internally to write the output tape were programmed by Mr. J. R. Ward of the USNPGS for use with a CalComp 165 graph-plotting routine, DRAW.

Figure 3 is an actual contour map from which approximately 1200 input points have been read. A polynomial of 400 terms was fitted and the elevations of a grid (spacing of 50 meters) were computed to plot Figure 4. The coefficients of this polynomial ranged from 904.18 for the constant term to -4.593×10^{-94} for the $x^{19}y^{19}$ term.

An analysis of a table of differences showed the maximum difference of 625 actual and computed elevations to be 31 feet in an elevation range of 900 to 1200 feet.

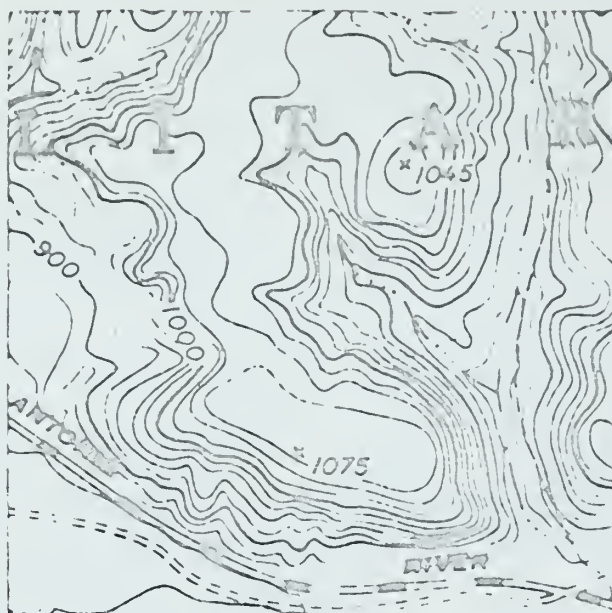


FIGURE 3

ACTUAL CONTOUR MAP OF GRID SQUARE 7174
U. S. ARMY HUNTER-LIGGETT MILITARY
RESERVATION

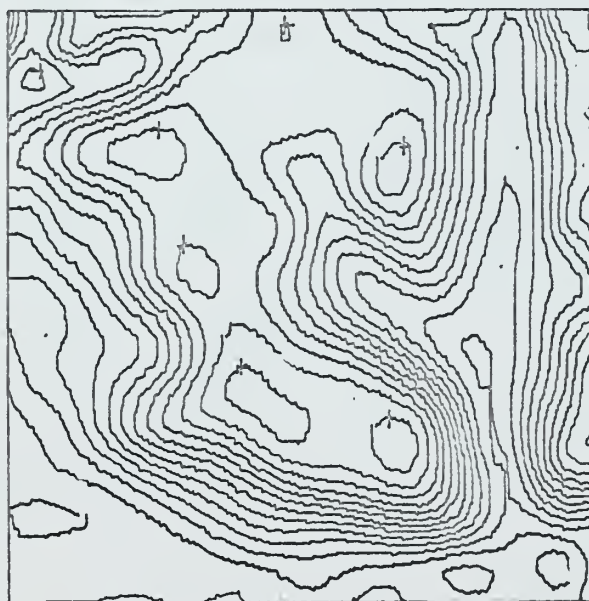


FIGURE 4

COMPUTER SIMULATION OF GRID SQUARE 7174
U. S. ARMY HUNTER-LIGGETT MILITARY
RESERVATION



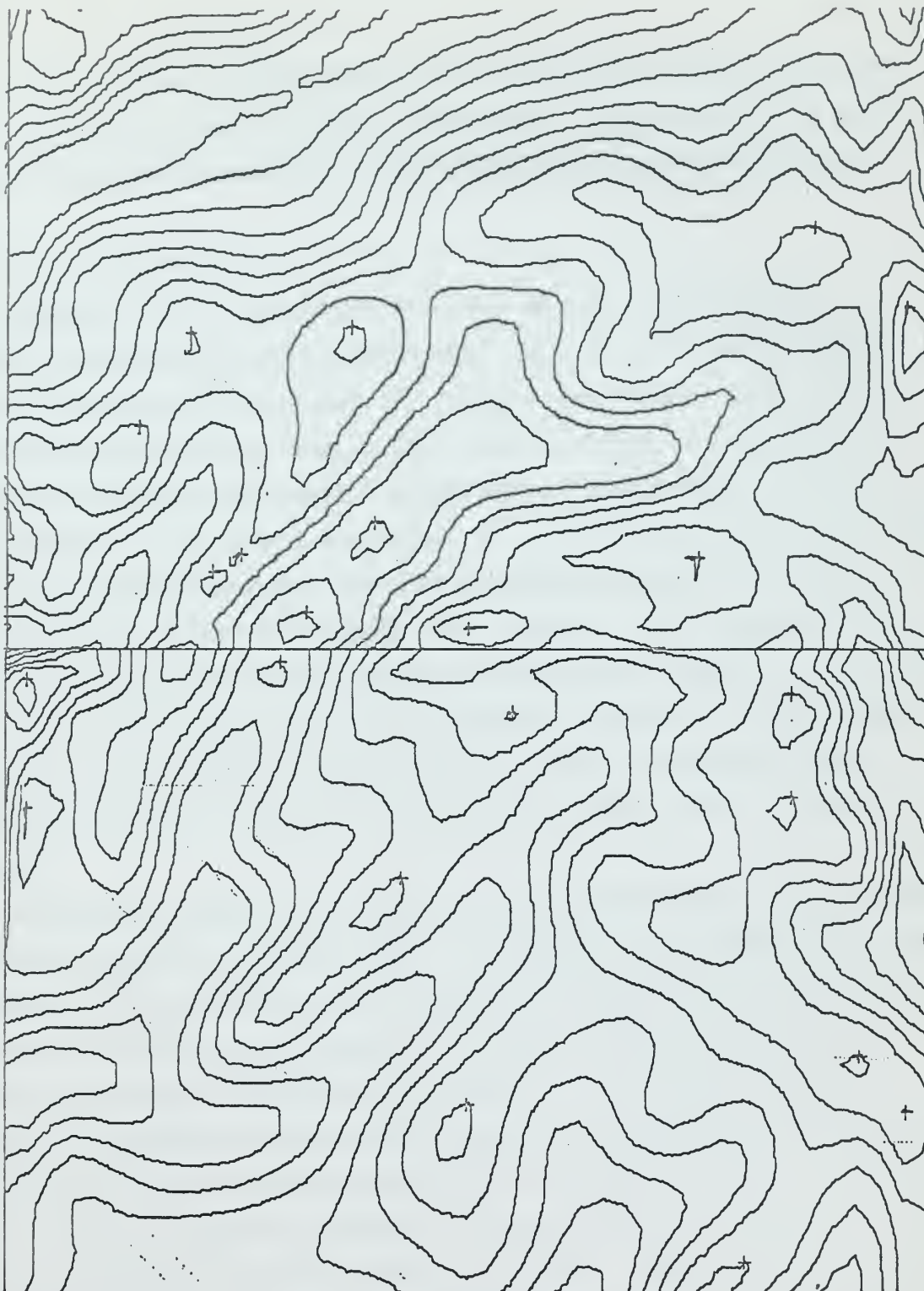


FIGURE 5

REPRESENTATIVE FIT AT THE JUNCTION
OF TWO GRID SQUARES



Figure 5 shows a representative fit at the junction of two of the grid squares approximated by separate polynomials. The accuracy with which the grid squares fit together is determined by the accuracy of the input data.

6. Analysis of the Model

PENAIR has been checked out with the use of a simple scenario. A 3000 x 4000 meter mountainous section of the U. S. Army Military Reservation was selected for simulation. Three 10-men enemy squads armed with M-14 rifles were situated as illustrated in Figure 6. A UH-1B helicopter was assigned the mission of delivering supplies from an airfield at position a to an outpost at position b. Two test runs were made with altitudes of the helicopter of 1000 feet and 500 feet above the terrain. In both tests all 18 evasive maneuvers were checked. Some representative flightpaths generated by the program are illustrated in Figures 6, 7, 8, and 9. The survival probabilities of the aircraft on the various attempts were computed and are shown in Tables 1 and 2. All of the computed probabilities are realistic and are, in a comparative sense, what one would expect.

A sensitivity analysis was then conducted on the rate of climb and descent and changes in aircraft speeds. A helicopter was simulated flying at 100 knots with its maximum rate of climb and descent taking the values of 500, 1000 and 2000 feet per minute. The effects of these three rates on nap of earth flight are illustrated in Figure 10. It should be noted that Figure 10 illustrates how the program generates the flightpath to overshoot the crest of a hill before descending just as a pilot tends to do. Figure 10 also shows, as should be expected, the greater the rate of climb and descent, the closer the aircraft remains to the earth. Figure 11 illustrates an aircraft flown with a maximum rate of climb and descent of 1500 feet per minute and speeds of 100, 200 and 300 knots. The resulting flightpaths also appear realistic.

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

The flightpath computed by PENAIR illustrated in Figure 12 is of a helicopter making a level turn at 70, 90 and 110 knots. Here it is noted, for the computed flightpath, the radius of turn increases as the airspeed increases.

The last analysis made was the comparison of computed results with actual results. Figure 13 is the simulated flightpath and terrain of an actual flightpath flown by a UH-1B over an actual terrain. The 44 weapon positions illustrated were actually manned by U. S. Army personnel. The observed time the helicopter was in view of each of these positions is listed in Table 3. The times in view for the simulated terrain and flightpath were computed and are also listed in Table 3. It is believed that with better terrain data an even closer comparison could be obtained.

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
530 CHICAGO HALL
CHICAGO, ILL. 60637

TO THE EDITOR:
I am writing to you to inform you of the results of my research on the effect of temperature on the rate of reaction of the following reaction:
$$2\text{H}_2\text{O}_2 \rightarrow 2\text{H}_2\text{O} + \text{O}_2$$

The reaction was carried out at various temperatures and the rate of reaction was measured. The results are shown in the table below:

Temperature (°C)	Rate of Reaction (mol/L·s)
10	0.0012
20	0.0025
30	0.0050
40	0.0100
50	0.0200

As you can see, the rate of reaction increases with increasing temperature. This is expected, as the rate of reaction is directly proportional to the temperature.

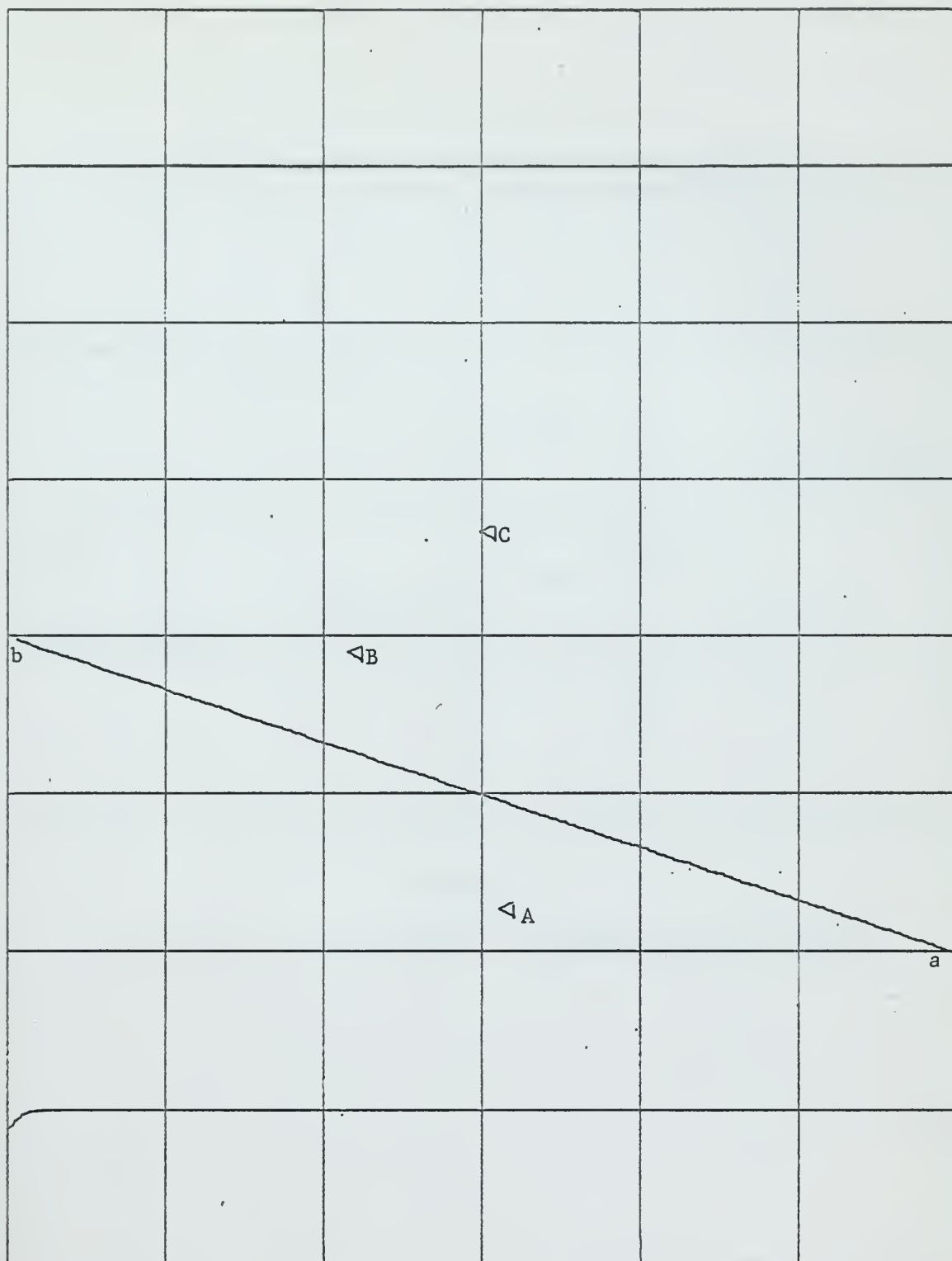


FIGURE 6

A STRAIGHT AND LEVEL FLIGHTPATH



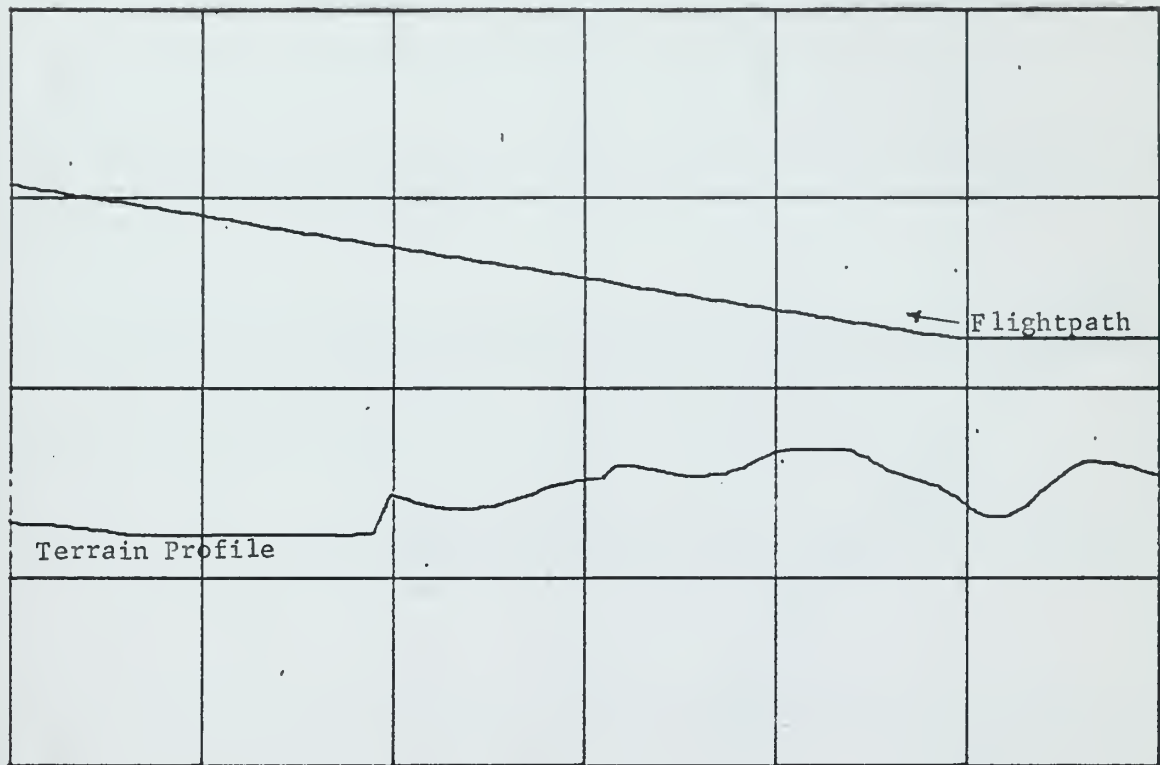


FIGURE 7

A FLIGHTPATH OF AN AIRCRAFT CLIMBING STRAIGHT AHEAD



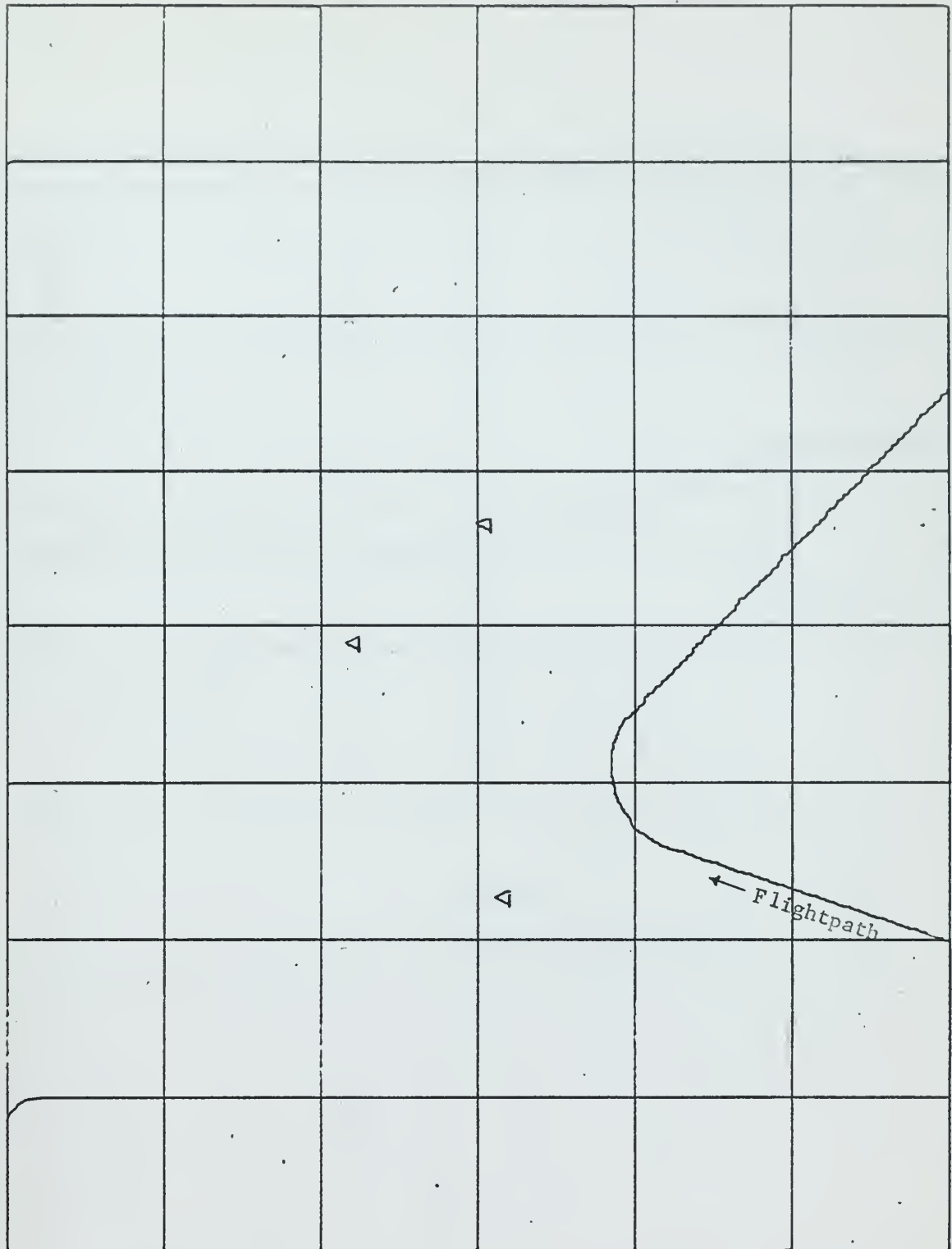


FIGURE 8

A FLIGHTPATH OF AN AIRCRAFT TURNING AWAY FROM A FIRING WEAPON



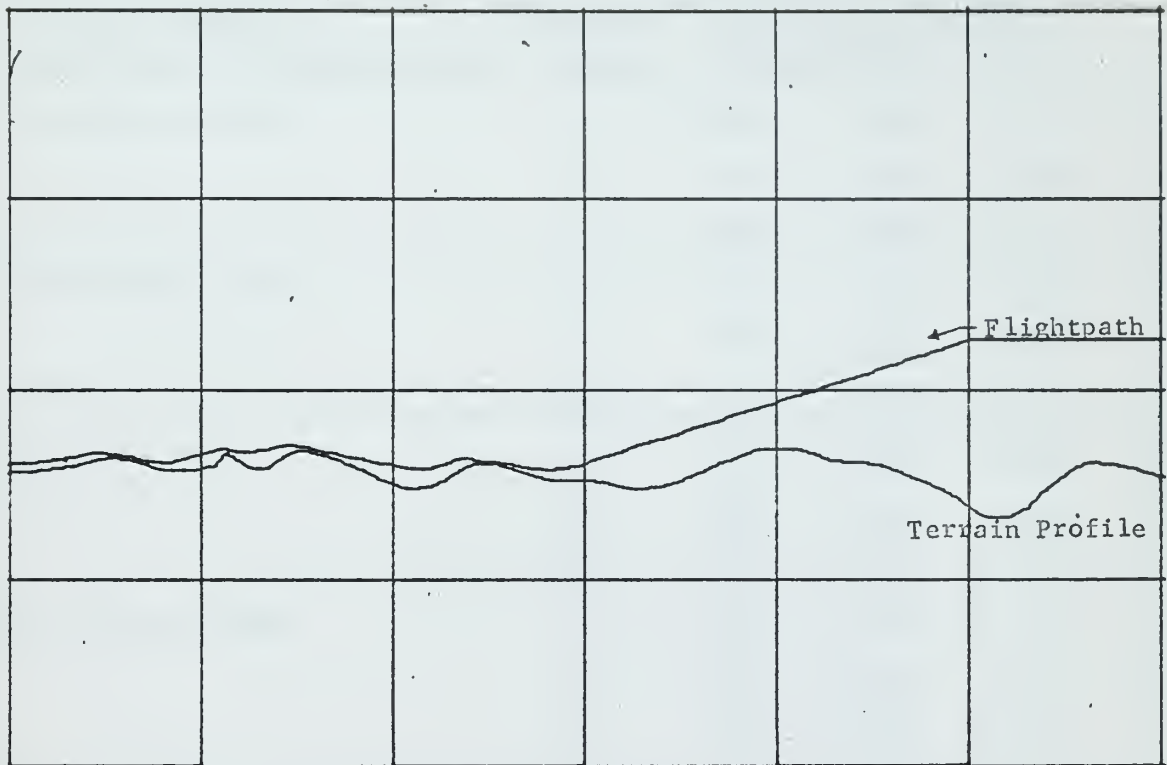


FIGURE 9

A NAP OF THE EARTH FLIGHTPATH



Page 1 of 1

Summary of Survival Probability of A UH-1B Helicopter Fired on by Three
10-Men Squads of M-14 Rifles

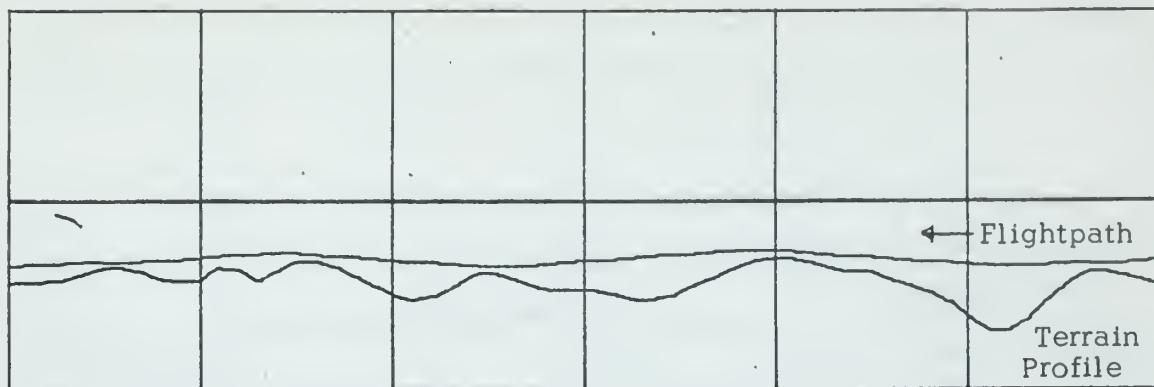
Initial Speed 90 Kts		Initial Altitude 650 Meters			
Evasive Maneuver (Commenced after receiving fire)	Evasive Speed	Survival Probabilities			Total
		Wpn A	Wpn B	Wpn C	
Fly straight and level	70	.8576	.8078	1.0000	.6928
	90	.8723	.8649	1.0000	.7545
	110	.8723	.8723	1.0000	.7610
Climb straight ahead	70	.8723	.8723	1.0000	.7610
	90	.8723	.8723	1.0000	.7610
	110	.8723	.8723	1.0000	.7610
Turn away, level	70	.9026	1.0000	1.0000	.9026
	90	.9026	1.0000	1.0000	.9026
	110	.9026	1.0000	1.0000	.9026
Turn away and climb	70	.9026	1.0000	1.0000	.9026
	90	.9026	1.0000	1.0000	.9026
	110	.9026	1.0000	1.0000	.9026
Dive and fly NOE	70	.8359	.7252	1.0000	.6062
	90	.8723	.8031	1.0000	.7005
	110	.8723	.8242	1.0000	.7190
Turn away and fly NOE	70	.8798	1.0000	1.0000	.8798
	90	.8874	1.0000	1.0000	.8874
	110	.8874	1.0000	1.0000	.8874

Table 1

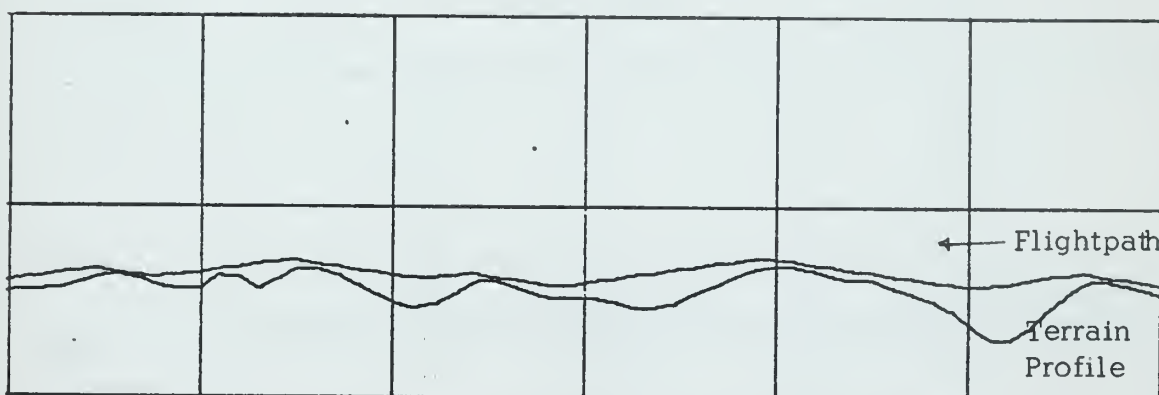
Summary of Survival Probabilities of UH-1B Helicopter Fired on by Three
10-Men Squads Armed with M-14 Rifles

Initial Speed 90 Kts	Initial Altitude 450 Meters				
Evasive Maneuver (Commenced after receiving fire)	Evasive Speed	Survival Wpn A	Wpn B	Probabilities Wpn C	Total
Fly straight and level	70	Values not determined at these speeds			
	90				
	110	.8309	.7788	1.0000	.6471
Climb straight ahead	70	Values not determined at these speeds			
	90				
	110	.8377	.8242	1.0000	.6905
Turn away, level	70	.8874	1.0000	1.0000	.8874
	90	.8874	1.0000	1.0000	.8874
	110	.8874	1.0000	1.0000	.8874
Turn away and climb	70	.8798	1.0000	1.0000	.8798
	90	.8798	1.0000	1.0000	.8798
	110	.8798	1.0000	1.0000	.8798
Dive and fly NOE	70	.7558	.9501	1.0000	.7181
	90	.8027	.9664	1.0000	.7758
	110	.3306	1.0000	1.0000	.8306
Turn away and fly NOE	70	.8798	1.0000	1.0000	.8798
	90	.8798	1.0000	1.0000	.8798
	110	.8798	1.0000	1.0000	.8798

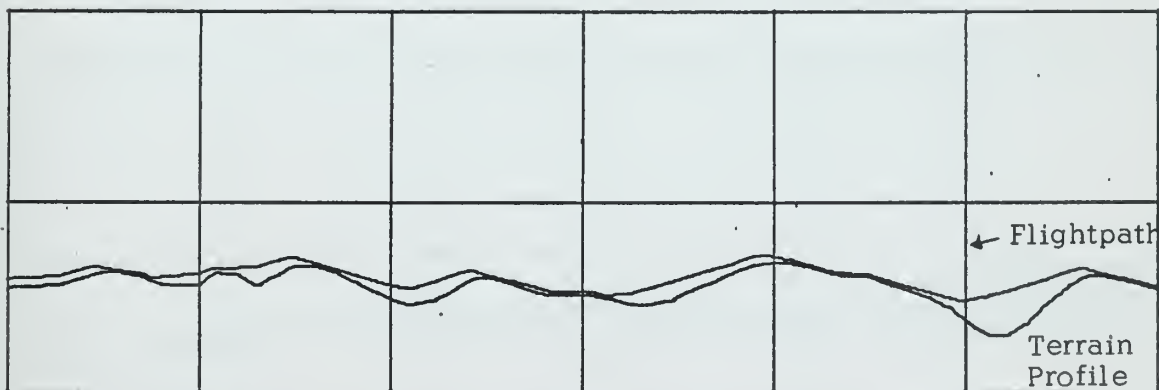
Table 2



RATE OF CLIMB = RATE OF DESCENT = 500 FEET PER MINUTE



RATE OF CLIMB = RATE OF DESCENT = 1000 FEET PER MINUTE



RATE OF CLIMB = RATE OF DESCENT = 2000 FEET PER MINUTE

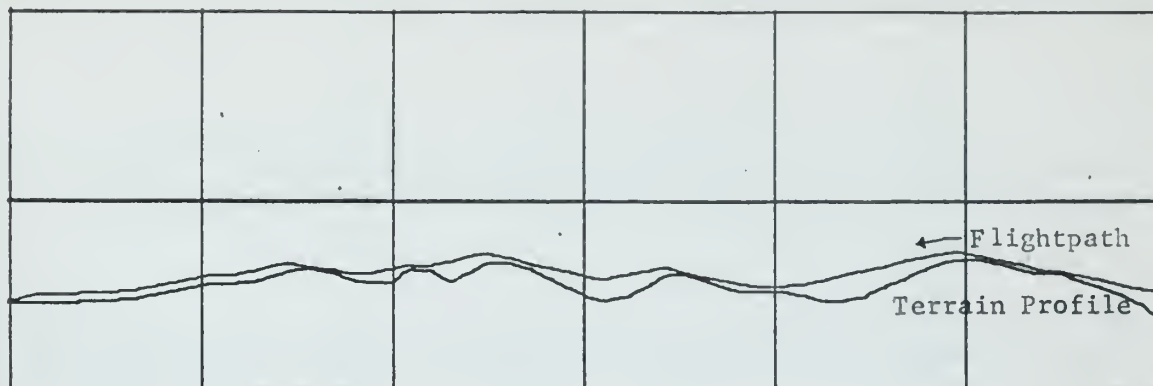
FIGURE 10

COMPARISON OF RATES OF CLIMBS AND DESCENTS IN A NAP
OF THE EARTH FLIGHT AT A CONSTANT SPEED OF 100 KTS.

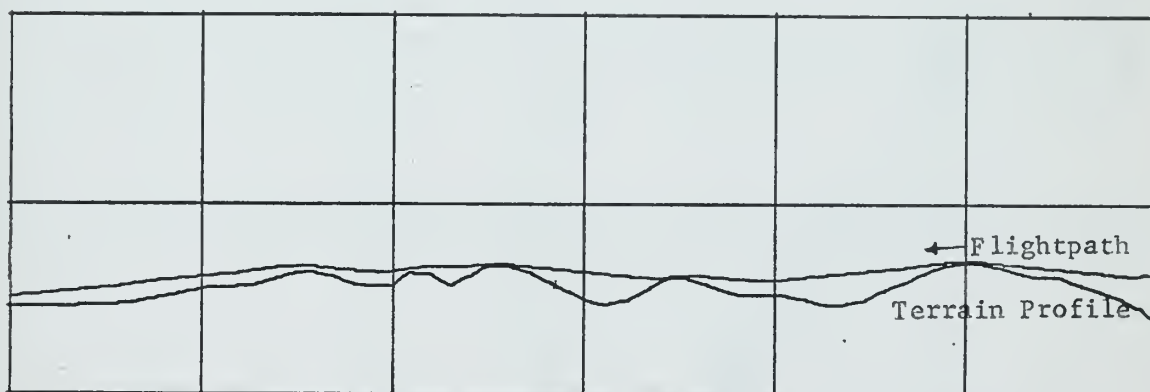
1. The first part of the paper discusses the importance of maintaining accurate records of all transactions. It emphasizes that proper record-keeping is essential for the integrity of the financial system and for the ability to detect and prevent fraud.

2. The second part of the paper examines the various methods used to collect and analyze data. It compares different techniques, such as direct observation, interviews, and the use of statistical models, and discusses the strengths and weaknesses of each. The author also discusses the importance of ensuring that the data is reliable and valid.

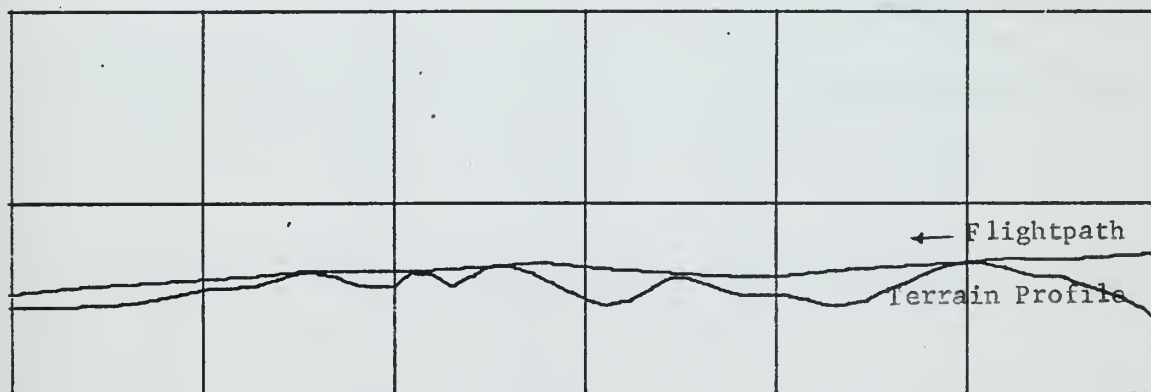
3. The third part of the paper discusses the results of the study. It presents the findings of the data analysis and discusses the implications of the results. The author concludes that the study has provided valuable insights into the behavior of the system and that the findings can be used to improve the system's performance and to prevent future problems.



100 Kts



200 KTS.



300 KTS.

FIGURE 11.

COMPARISON OF AIRCRAFT SPEEDS ON A MAP OF THE EARTH FLIGHT

1. The first part of the paper discusses the importance of maintaining accurate records of all transactions. This is essential for the proper management of the company's finances and for ensuring that all transactions are properly documented and audited.

2. The second part of the paper discusses the importance of maintaining accurate records of all transactions. This is essential for the proper management of the company's finances and for ensuring that all transactions are properly documented and audited.

3. The third part of the paper discusses the importance of maintaining accurate records of all transactions. This is essential for the proper management of the company's finances and for ensuring that all transactions are properly documented and audited.

4. The fourth part of the paper discusses the importance of maintaining accurate records of all transactions. This is essential for the proper management of the company's finances and for ensuring that all transactions are properly documented and audited.

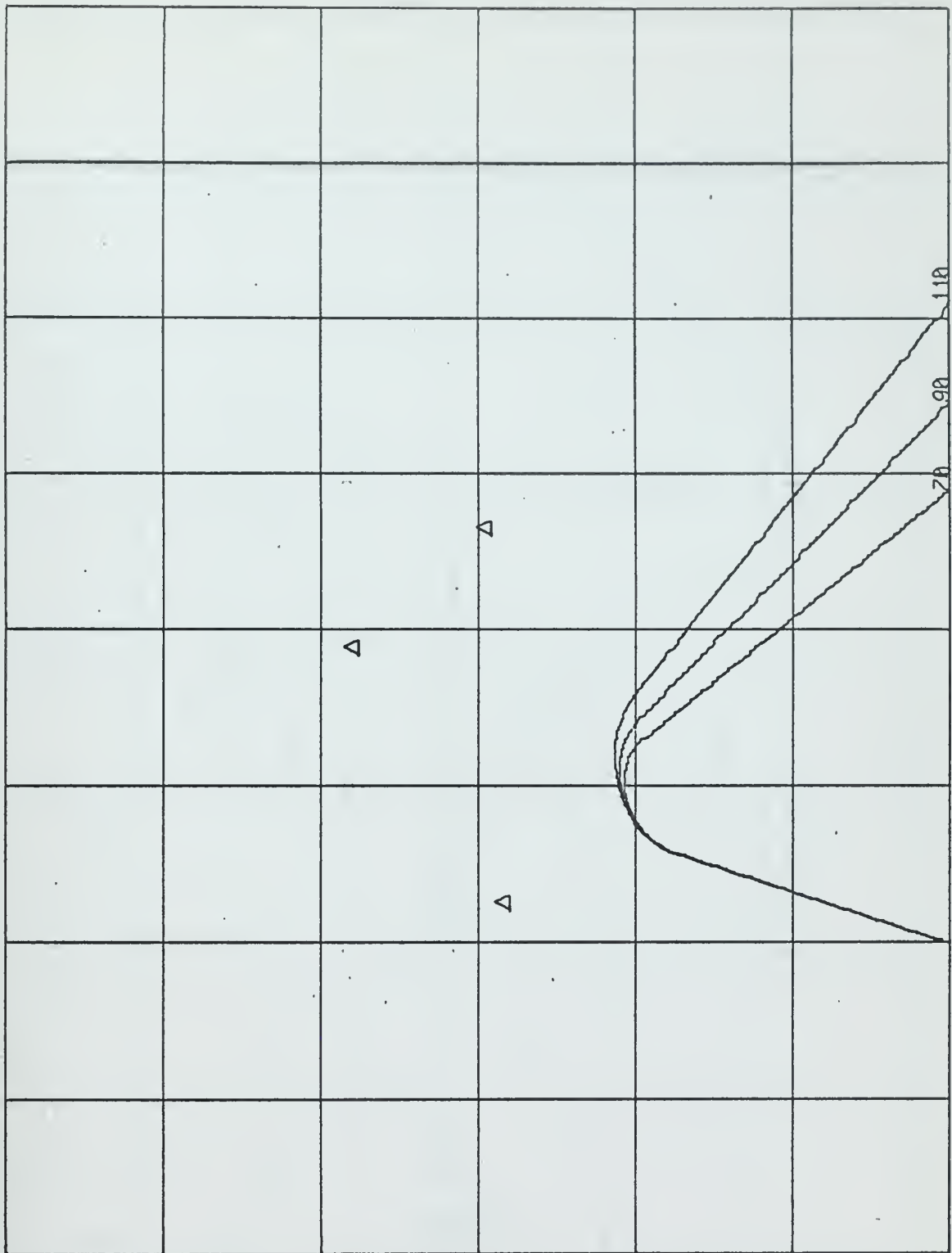


FIGURE 12

COMPARISON OF THE AIRCRAFT'S RADIUS OF TURN
AT 3 AIRSPEEDS, 70, 90 and 110 KTS.

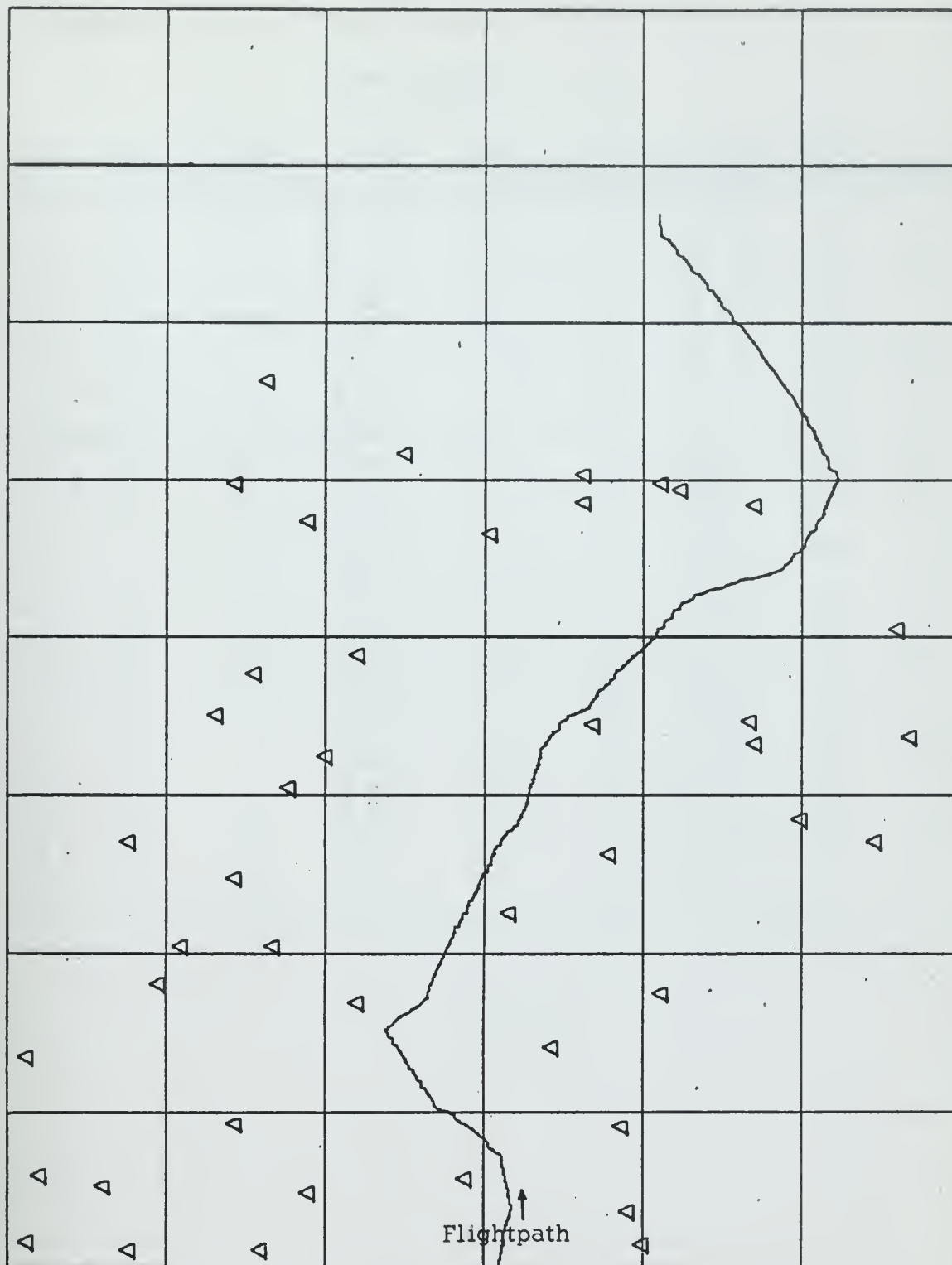


FIGURE 13

COMPUTER SIMULATION OF AN ACTUAL FLIGHTPATH
FLOWN BY A UH-1B HELICOPTER



Comparison of Actual and Computed Time Intervals in View of 44 Weapon Sites

Time Interval = .6 sec.

Wpn. No	Time Actual	in View Computed	Wpn No	Time Actual	In View Computed	Wpn No	Time Actual	In View Computed
1	0	0	16	50	56	31	0	0
2	0	9	17	0	0	32	22	39
3	30	30	18	43	47	33	0	0
4	0	0	19	63	108	34	89	88
5	0	0	20	4	30	35	93	91
6	0	59	21	17	74	36	69	76
7	0	0	22	0	72	37	66	58
8	0	40	23	0	0	38	49	64
9	0	0	24	0	0	39	16	36
10	0	17	25	59	63	40	0	16
11	9	2	26	0	34	41	8	8
12	0	0	27	63	64	42	56	56
13	26	39	28	43	45	43	19	15
14	40	63	29	0	9	44	103	102
15	55	73	30	0	0			

Table 3

7. Conclusions and recommendations.

The simulation of the terrain obtained by this program has proven to be extremely realistic. (See Figure 4, Section 5, page 11). The junctions of squares approximated by separate polynomials are as good as the data used for the approximation. (See Figure 5, Section 5, page 12). This program was written in its present form because of the availability of input data to the author in the format described in Appendix V. If the actual elevations could be obtained using an equally spaced grid for the x and y coordinates, the first overlay could be eliminated. It is felt that this is feasible and would make the preparation of the input cards much simpler.

Among the uses of the simulation of the terrain by this method include ground to ground combat and ground to air combat environments. Mask angles, terrain profiles and actual distances between points in a mountainous terrain are all easily determined. The uses of the PENAIR program include the determination of optimal flightpaths across hostile terrain, the survival probability and optimal maneuvers to perform if the aircraft is fired upon.

The results in Section 6 all tend to illustrate the realism of the aircraft penetration model. Whether the actual kill probabilities used are exact is not extremely important since an appropriate defensive maneuver for an unarmed aircraft to initiate upon receiving ground fire may be obtained by observing the relative survival probabilities of several maneuvers.

It is believed that with additional programming this model could be extended to include formations of two or more aircraft, thus greatly increasing its usefulness. In addition, since the terrain is simulated by mathematical functions and since the line of sight between aircraft and weapon may be described by a linear relationship, it appears possible to solve the two relations for a point of intersection. Upon accomplishing this programming, it is believed the computer running time would be reduced significantly.

BIBLIOGRAPHY

1. Graybill, F. A. An introduction to linear statistical models, volume 1. McGraw-Hill Book Company, Inc., 1961.
2. Clark, R. E., Kubik, R. N., and Phillips, L.P. Orthogonal polynomial least squares surface fit. Communications of the ACM, v. 6, April, 1963: 162-163.
3. Dayhoff, M.O. A contour-map program for x-ray crystallography. Communications of the ACM, v. 6, October, 1963: 620-622.

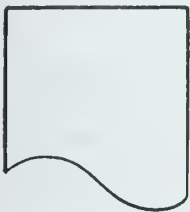
APPENDIX I

Flow Charts of the PENAIR Program

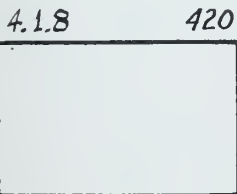
In the flow charts of this appendix, the following conventions are used:



This symbol is used for a READ statement.



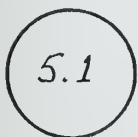
This symbol is used for a PRINT statement.



A rectangle indicates one or more FORTRAN statements. A number such as 4.1.8 indicates the statement in this rectangle is further explained by Flow Chart 4.1.8 in this appendix. A number such as 420 is the statement number as it appears in the FORTRAN listing in Appendix II.



This symbol is used for an IF statement.



This symbol indicates the logic of this Flow Chart is continued at the beginning of Flow Chart 5.1 of this appendix.

Section 1: Introduction

The purpose of this document is to provide a comprehensive overview of the project's objectives and scope.

The project aims to develop a new software application that will streamline the workflow of the department.

The project is divided into several phases, including planning, development, testing, and deployment.

The project team consists of several members, each with specific responsibilities and expertise.

The project is expected to be completed within a timeline of six months.

The project budget is estimated to be within the allocated funds.

The project will be managed using a structured approach to ensure successful outcomes.

The project will be subject to regular monitoring and reporting to the management team.

The project team will maintain open communication throughout the project lifecycle.

The project will be reviewed at regular intervals to assess progress and address any issues.

The project team will ensure that all project goals and objectives are met.

The project will be completed on time and within budget.

The project team will provide a final report detailing the project's outcomes and lessons learned.



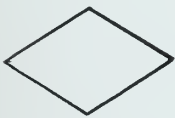
This symbol indicates a connector between two parts of the same Flow Chart of this appendix.



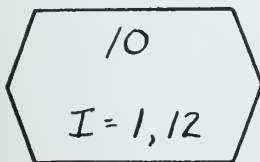
This symbol indicates the logic of this Flow Chart is continued where the same symbol appears in Flow Chart 4.1 of this appendix.



This is the symbol for a CONTINUE statement whose statement number is 10.

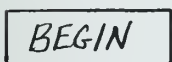


This is the symbol for a computed GO TO statement.

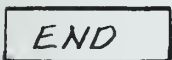


This symbol is used for the statement

DO 10 I = 1, 12



This symbol indicates the beginning or entry point for the Flow Chart.



This symbol indicates the end or exit point for the Flow Chart.

1. The first part of the paper discusses the importance of the study and the objectives of the research.

2. The second part of the paper describes the methodology used in the study and the data collection process.

3. The third part of the paper presents the results of the study and discusses the findings in detail.

4. The fourth part of the paper discusses the implications of the study and provides recommendations for future research.

5. The fifth part of the paper concludes the study and summarizes the main findings and conclusions.

6. The sixth part of the paper provides a list of references and sources used in the study.

1. The first part of the paper discusses the importance of the study and the objectives of the research.

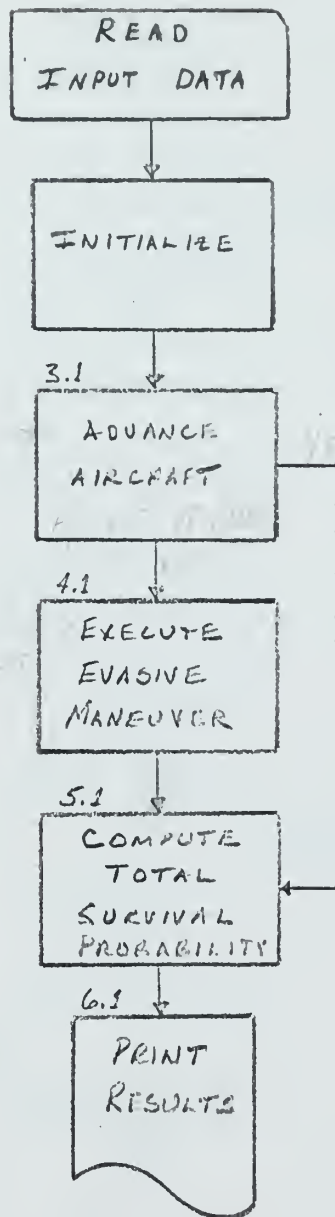
2. The second part of the paper describes the methodology used in the study and the data collection process.

3. The third part of the paper presents the results of the study and discusses the findings in detail.

4. The fourth part of the paper discusses the implications of the study and provides recommendations for future research.

5. The fifth part of the paper concludes the study and summarizes the main findings and conclusions.

6. The sixth part of the paper provides a list of references and sources used in the study.

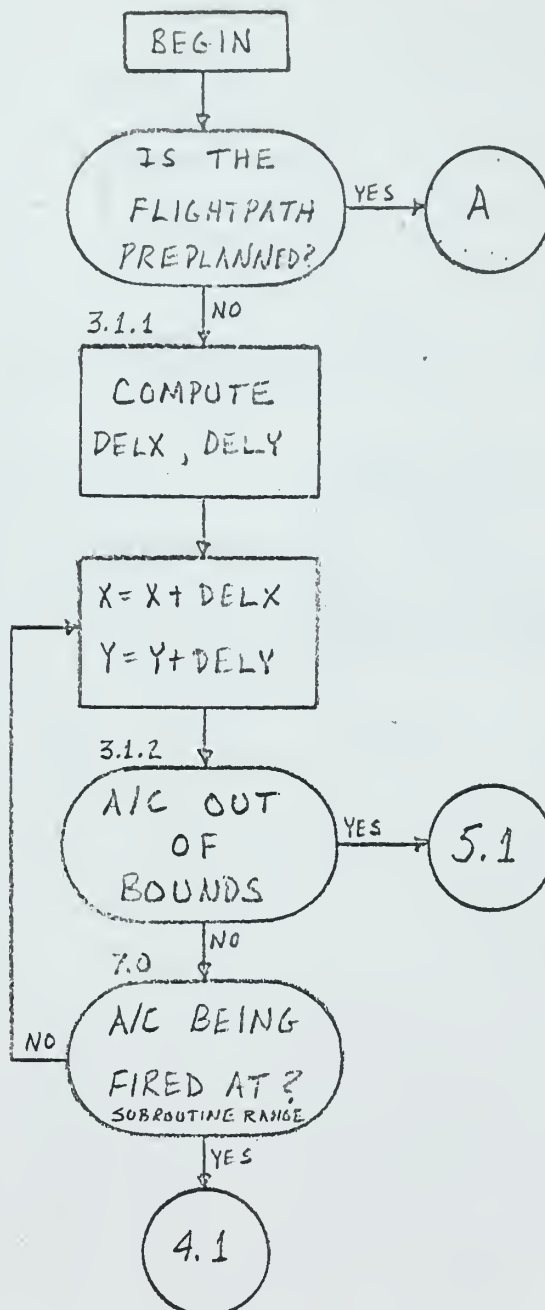


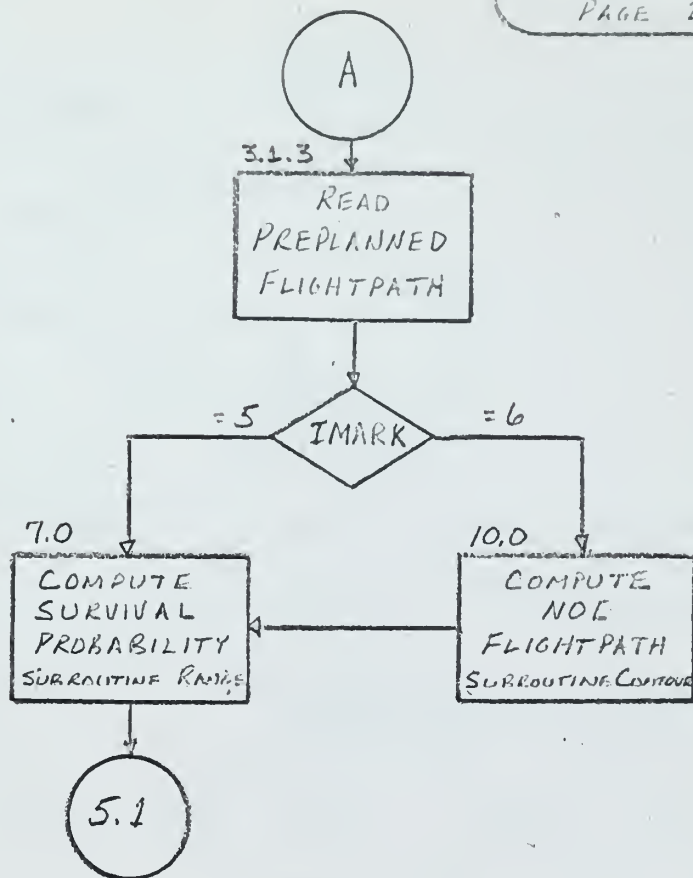
GENERAL FLOW CHART ILLUSTRATING LOGICAL PATHS OF PENAIR.

3.1

ADVANCE
AIRCRAFT

FLOWCHART 3.1
PAGE 1 OF 2

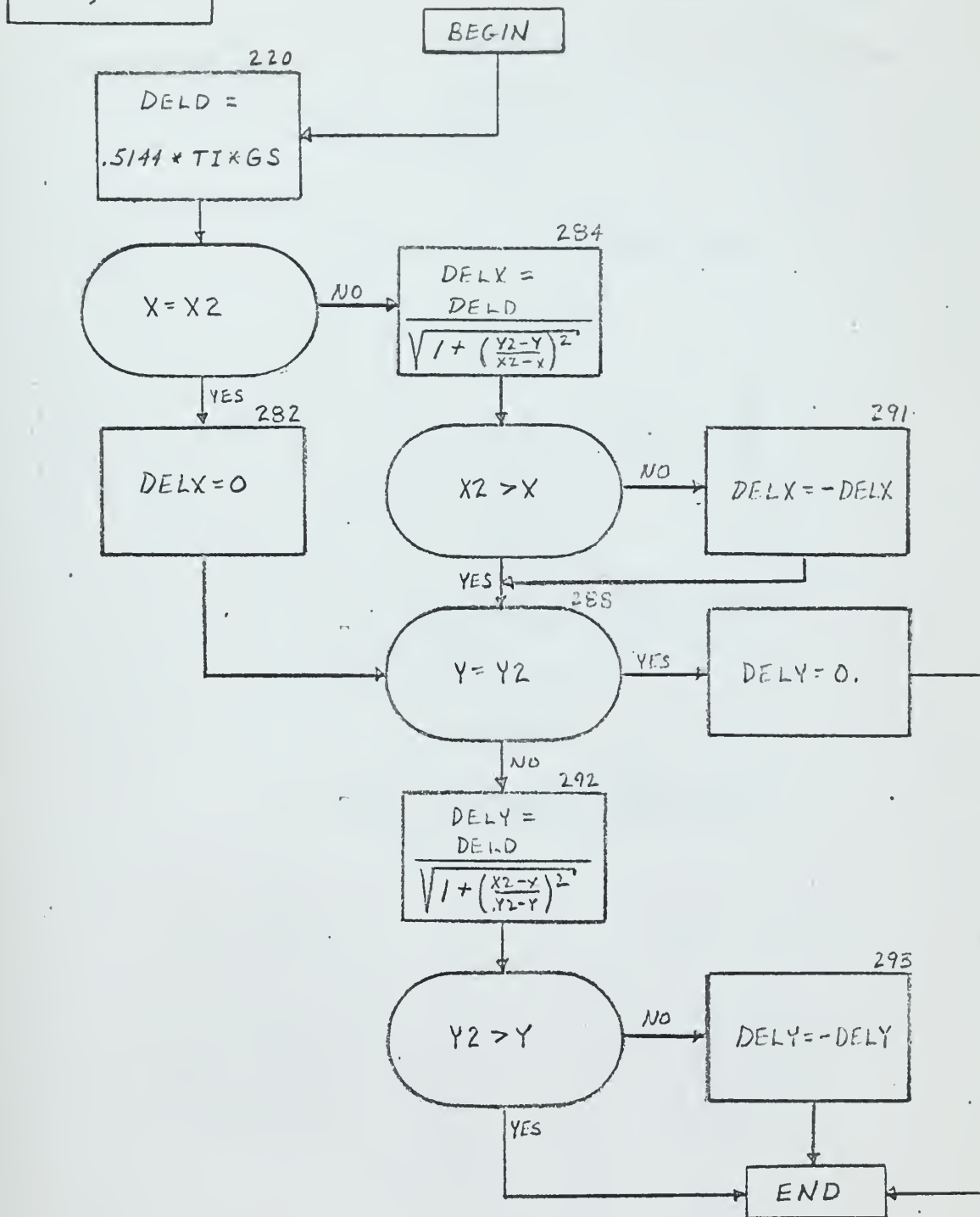




3.1.1

COMPUTE
DELX, DELY

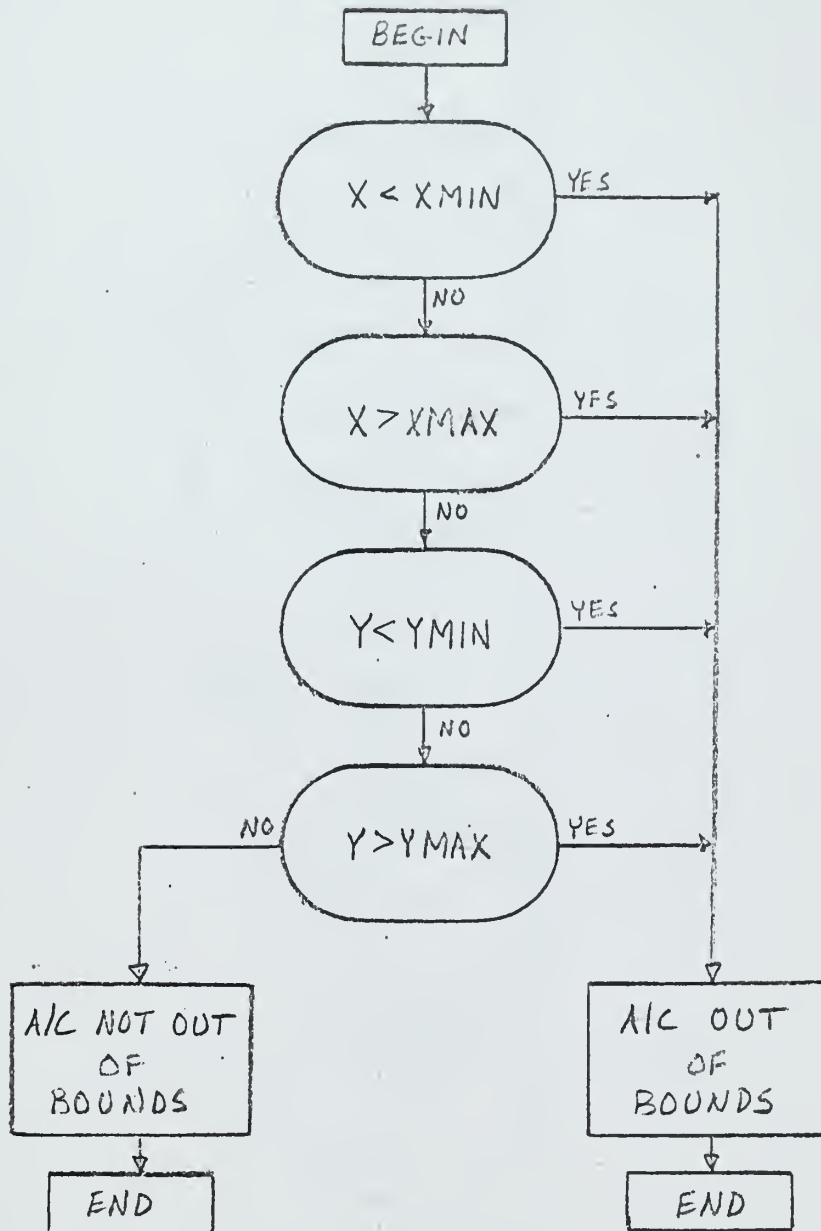
FLOW CHART 3.1.1
PAGE 1 OF 1



3.1.2

A/C OUT OF
BOUNDS ?

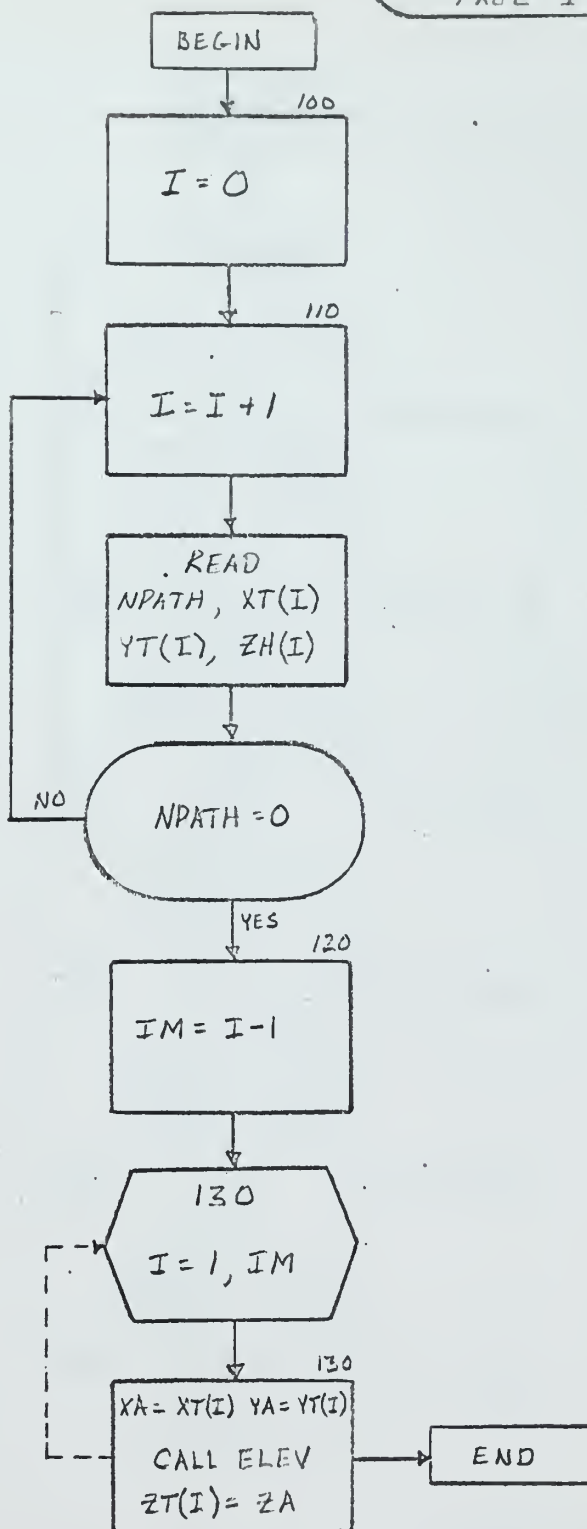
FLOW CHART 3.1.2
PAGE 1 OF 1

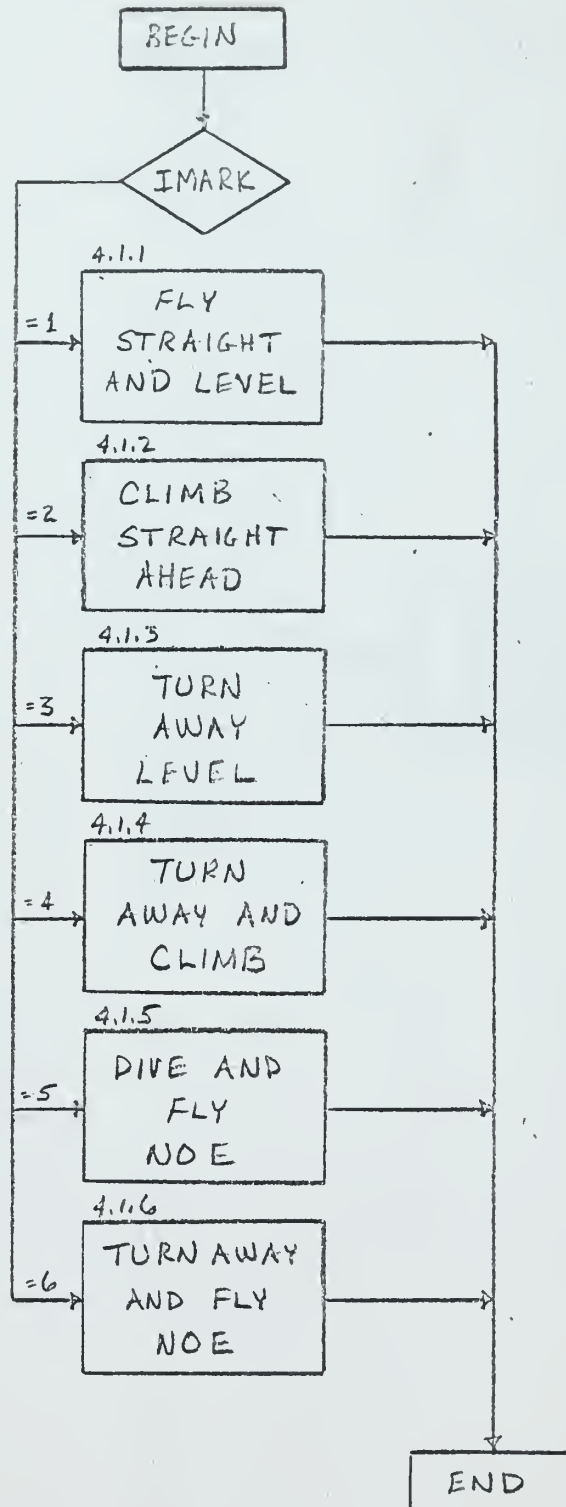


3.1.3

READ
PREPLANNED
FLIGHTPATH

FLOW CHART 3.1.3
PAGE 1 OF 4

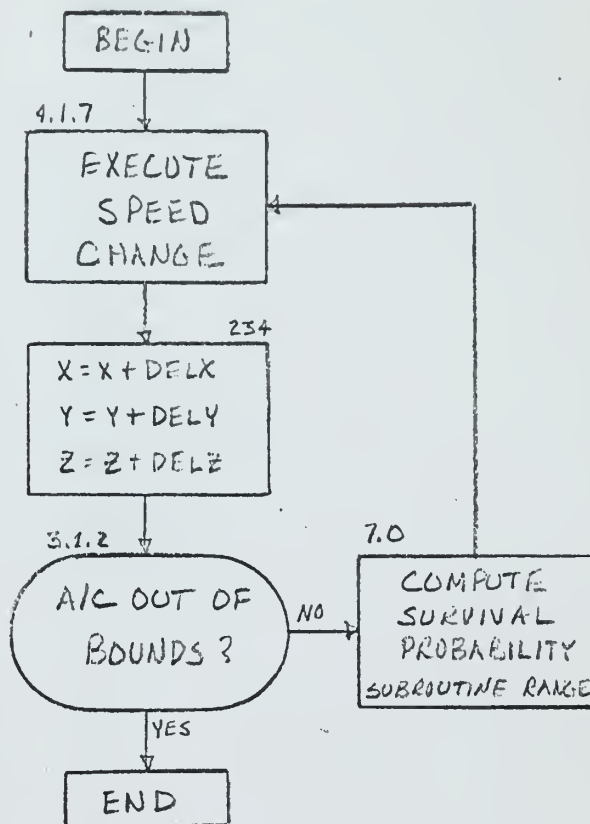


EXECUTE
EVASIVE
MANEUVERFLOW CHART 4.1
PAGE 1 OF 1

4.1.1

FLY
STRAIGHT
AND LEVEL

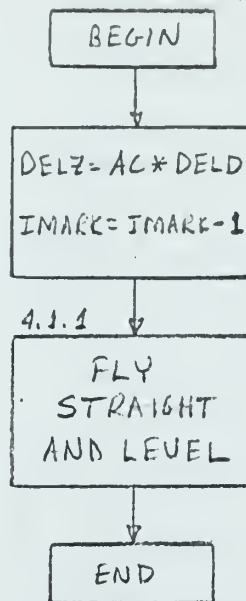
FLOW CHART 4.1.1
PAGE 1 OF 1



4.1.2

CLIMB
STRAIGHT
AHEAD

FLOW CHART 4.1.2
PAGE 1 OF 1



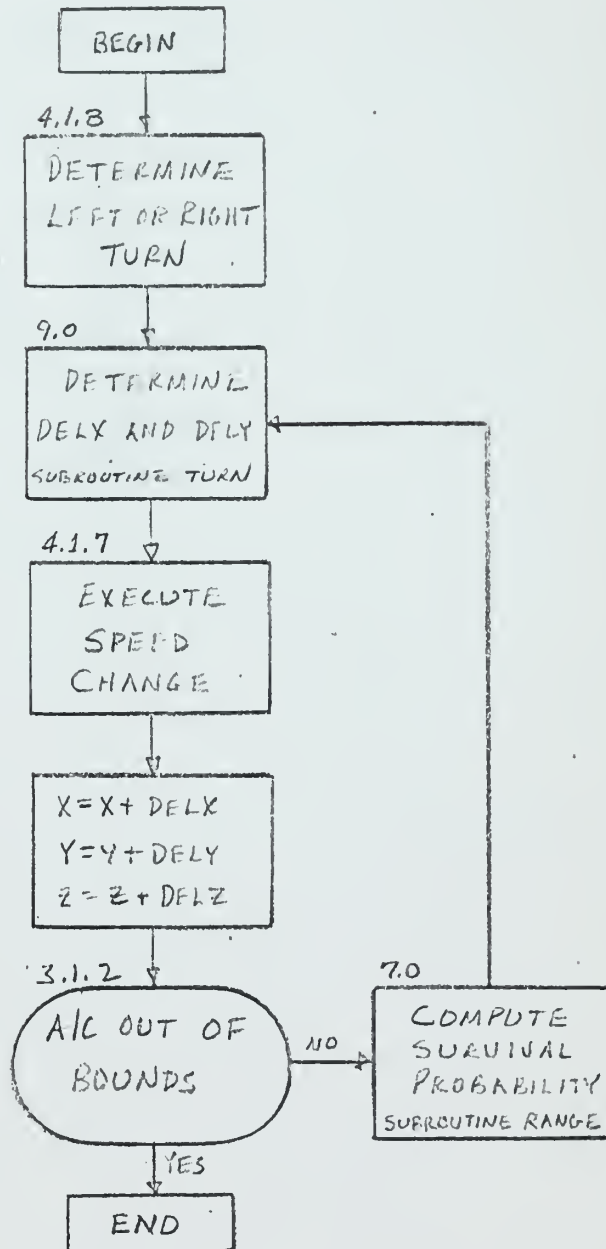


4.1.3

TURN
AWAY
LEVEL

Flow Chart 4.1.3

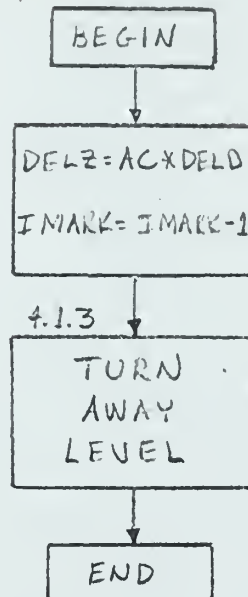
PAGE 1 OF 1



4.1.4

TURN
AWAY AND
CLIMB

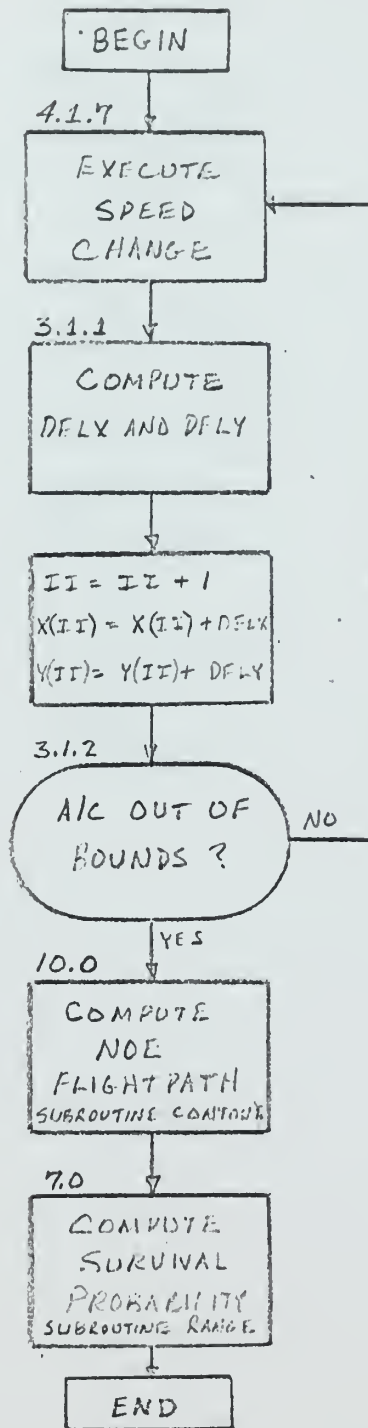
FLOW CHART 4.1.4
PAGE 1 OF 1



4.1.5

DIVE AND
FLY
NOE

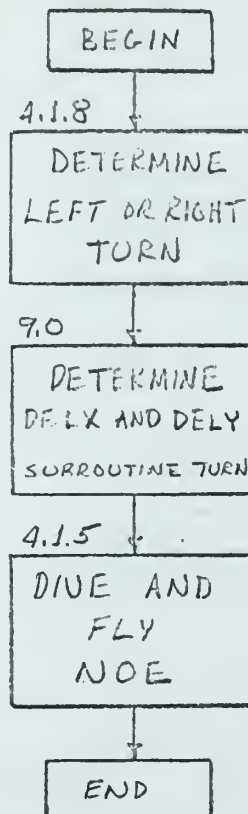
FLOW CHART 4.1.5
PAGE 1 OF 1



4.1.6

TURN AWAY
AND FLY
NOE

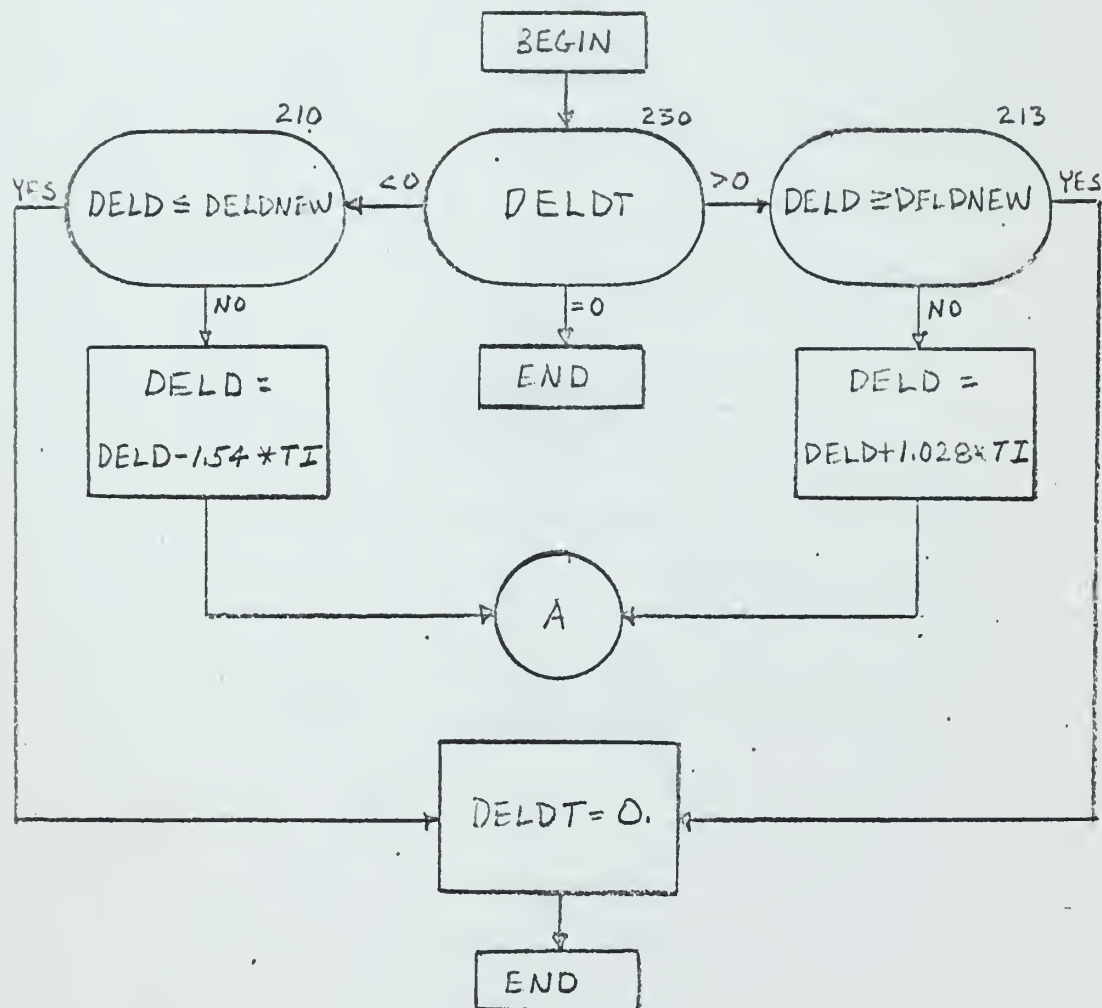
FLOW CHART 4.1.6
PAGE 1 OF 1

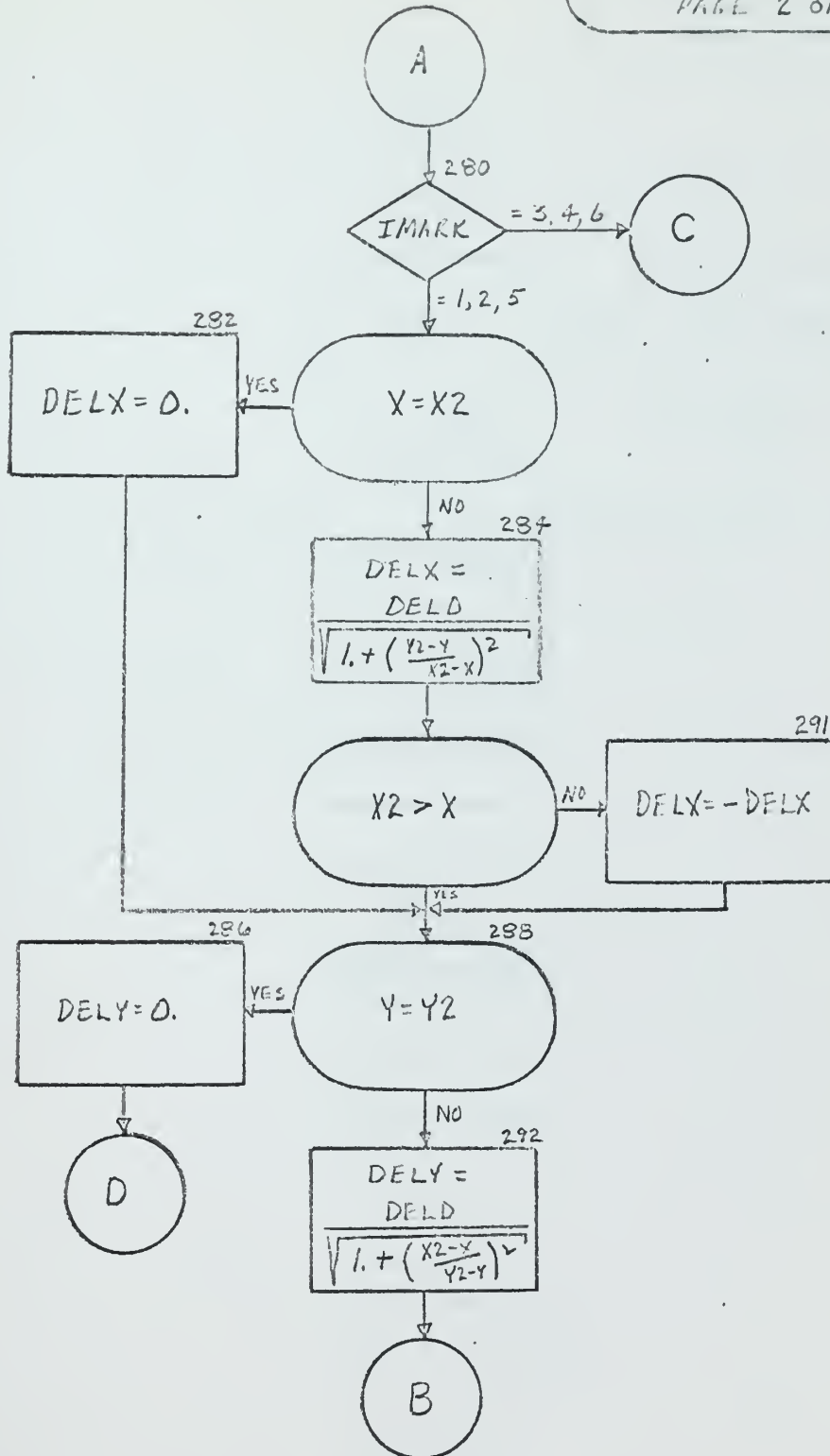


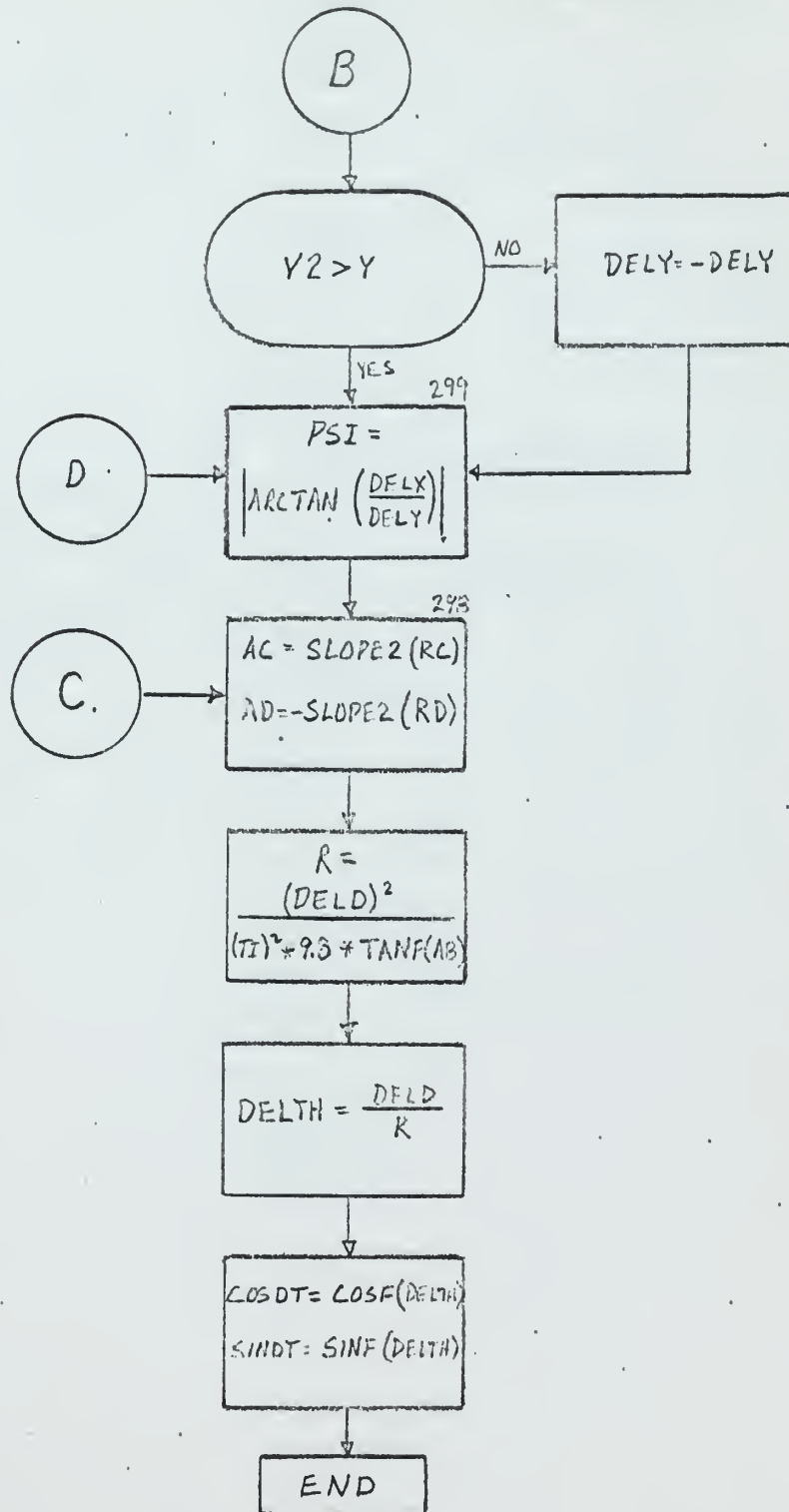
4.1.7

EXECUTE
SPEED
CHANGE

FLOW CHART 4.1.7
PAGE 1 OF 3



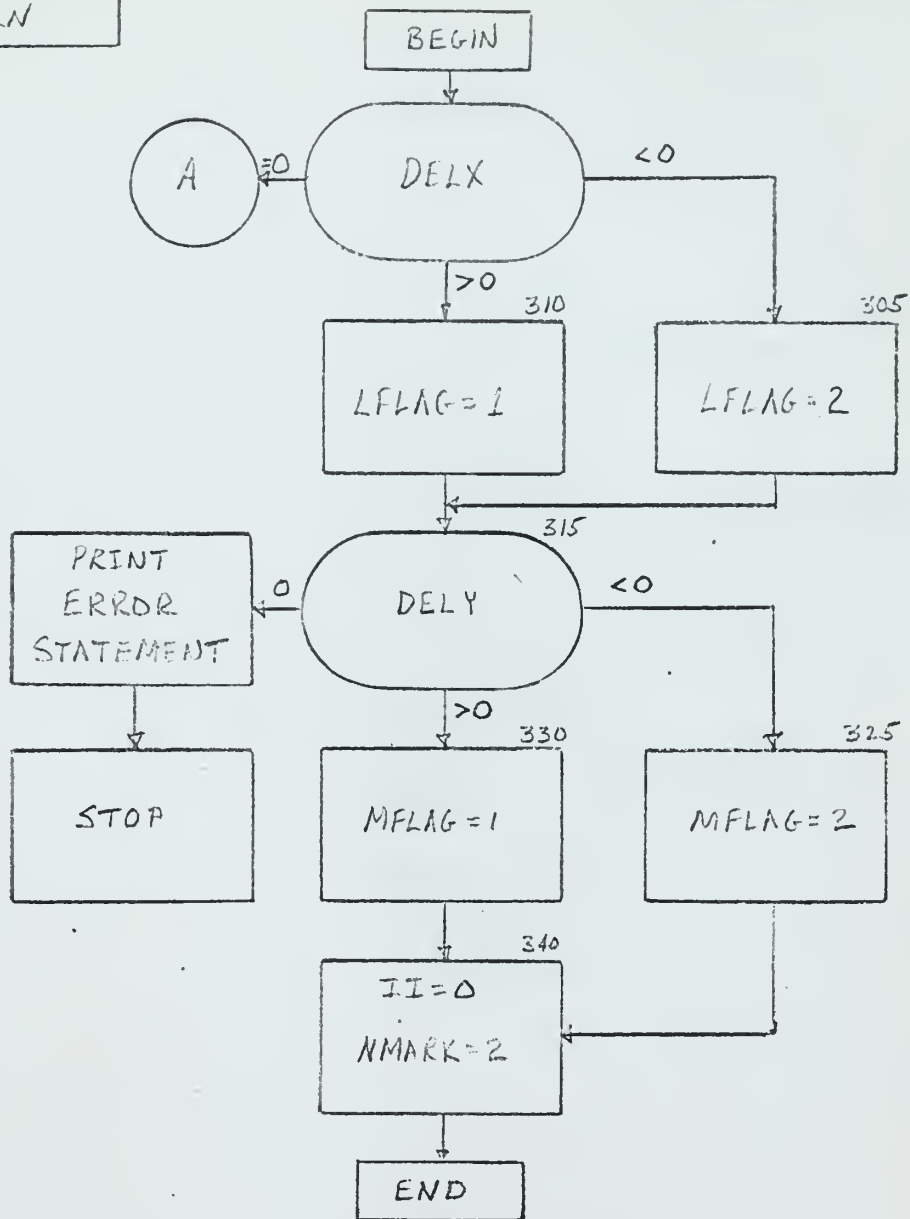


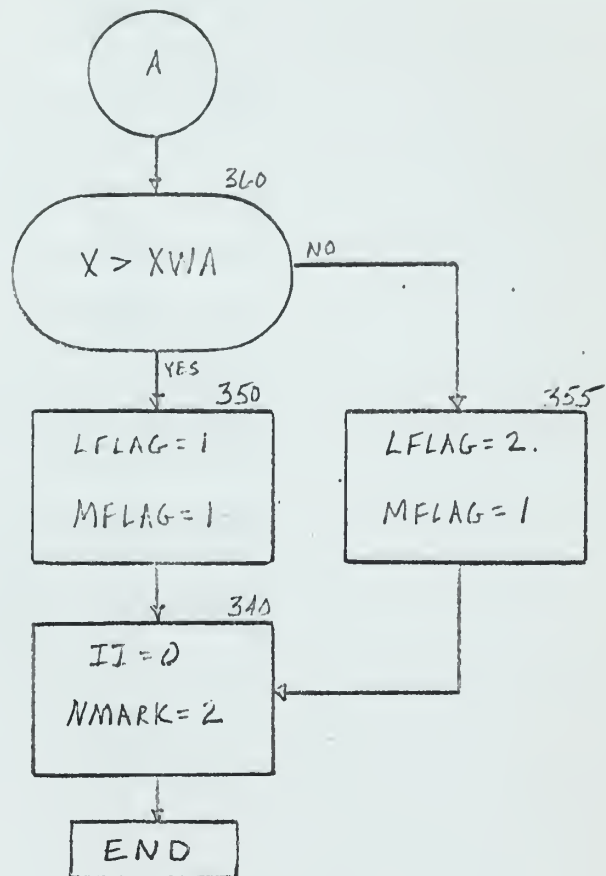


4.1.8

DETERMINE
LEFT OR RIGHT
TURN

FLOW CHART 4.1.8
PAGE 1 OF 2

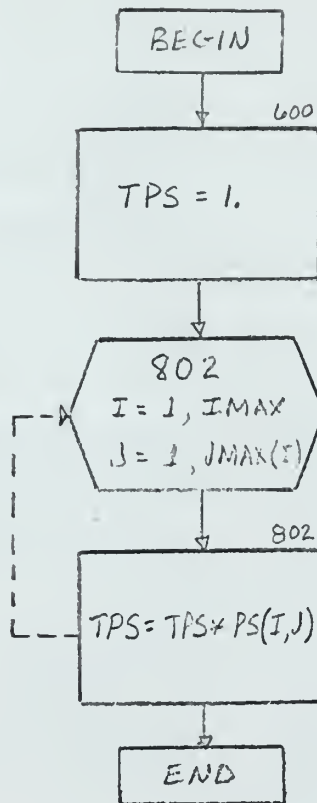




5.1

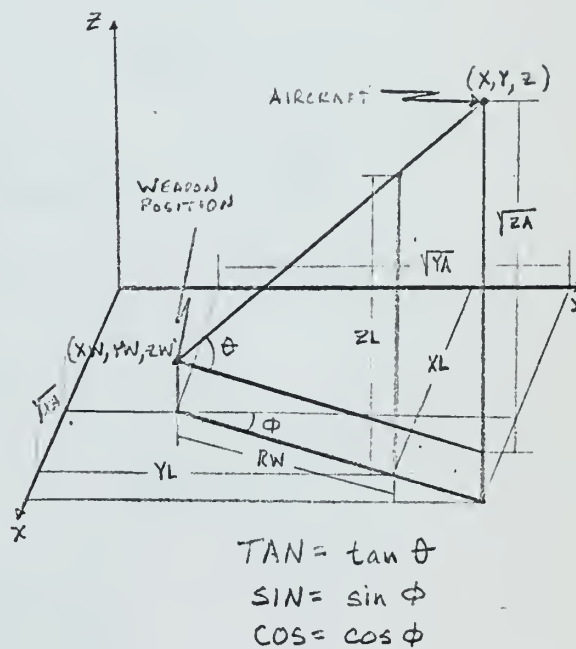
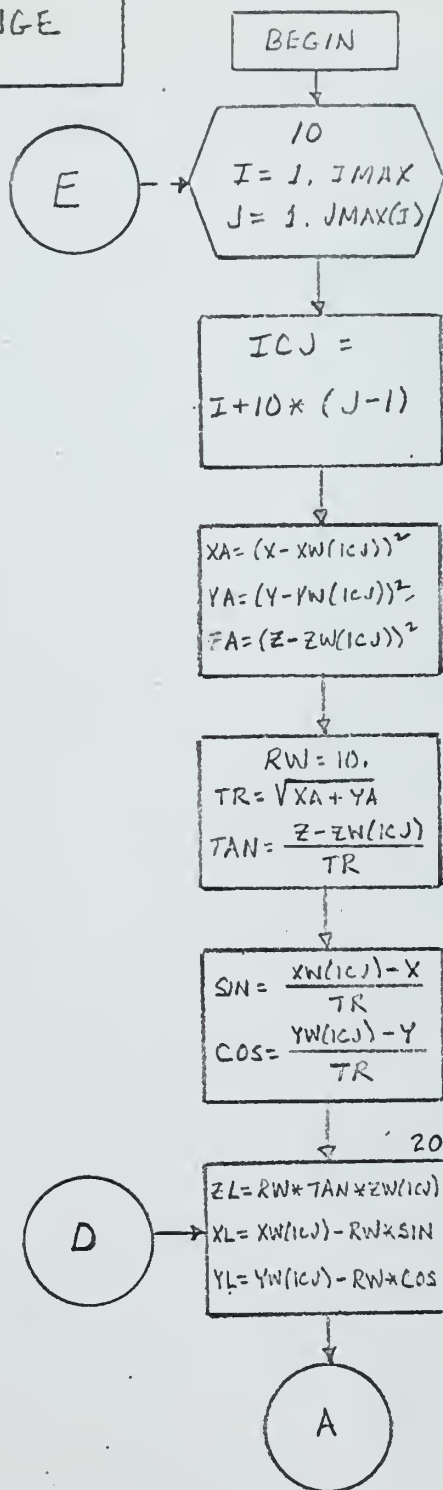
COMPUTE
TOTAL
SURVIVAL
PROBABILITY

FLOW CHART 5.1
PAGE 1 OF 1

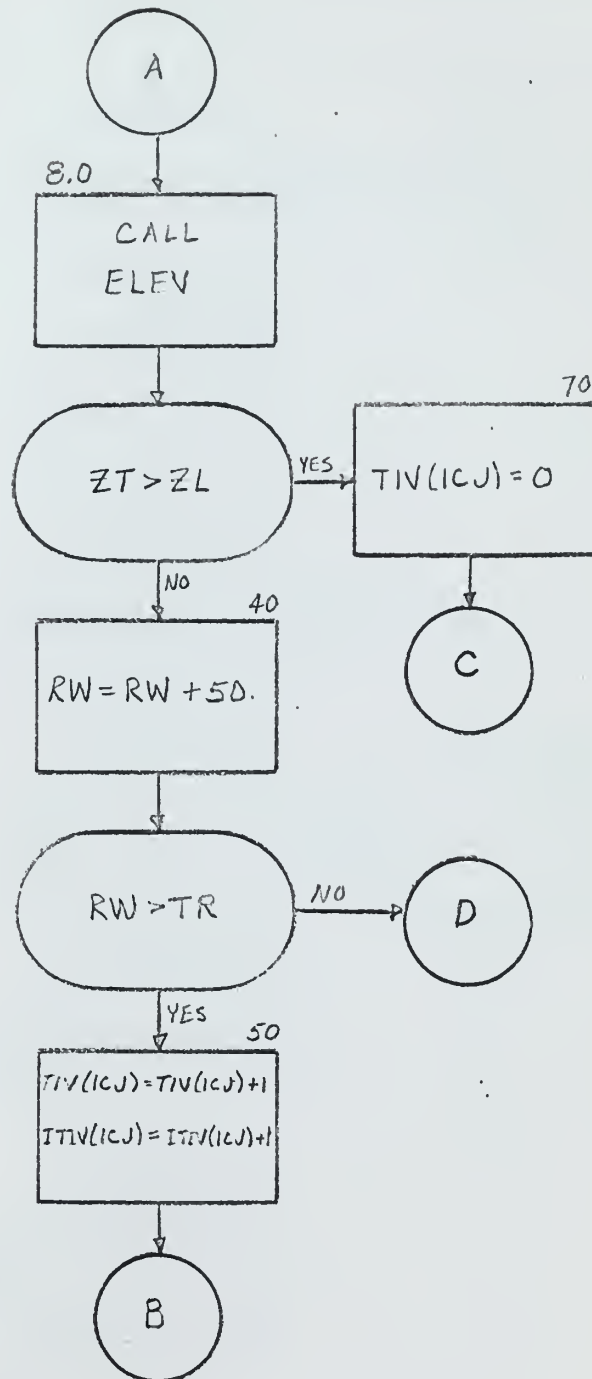


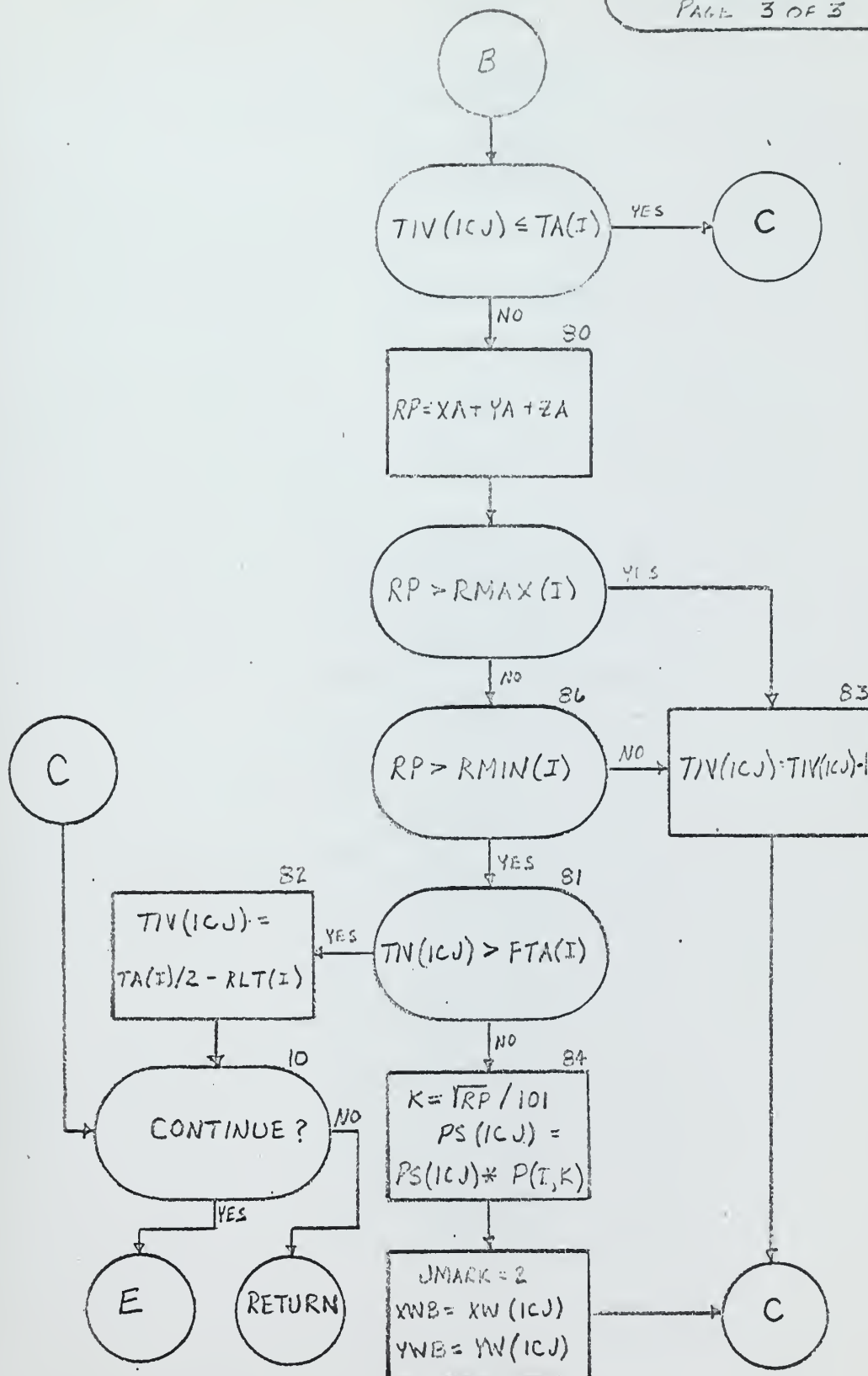


7.0

SUBROUTINE
RANGEFLOW CHART 7.0
PAGE 1 OF 3



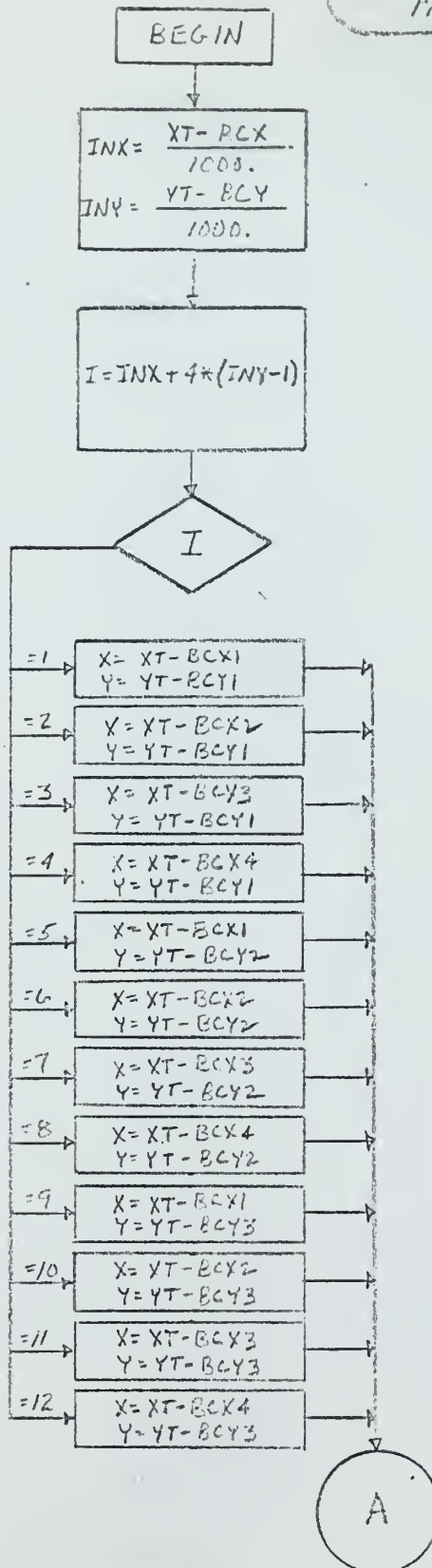


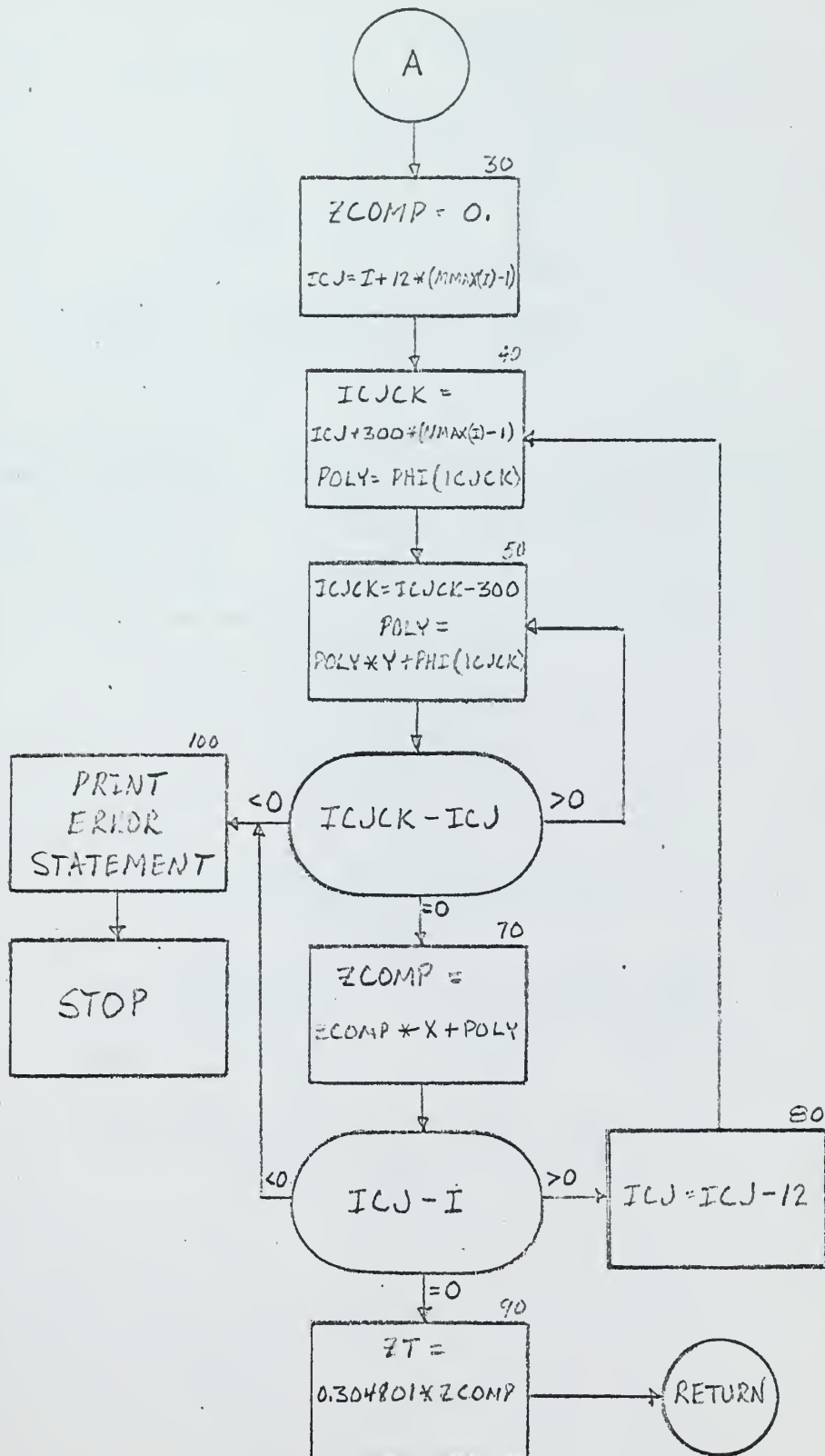


8.0

SUBROUTINE
ELEV

FLOW CHART 8.0
PAGE 1 OF 2

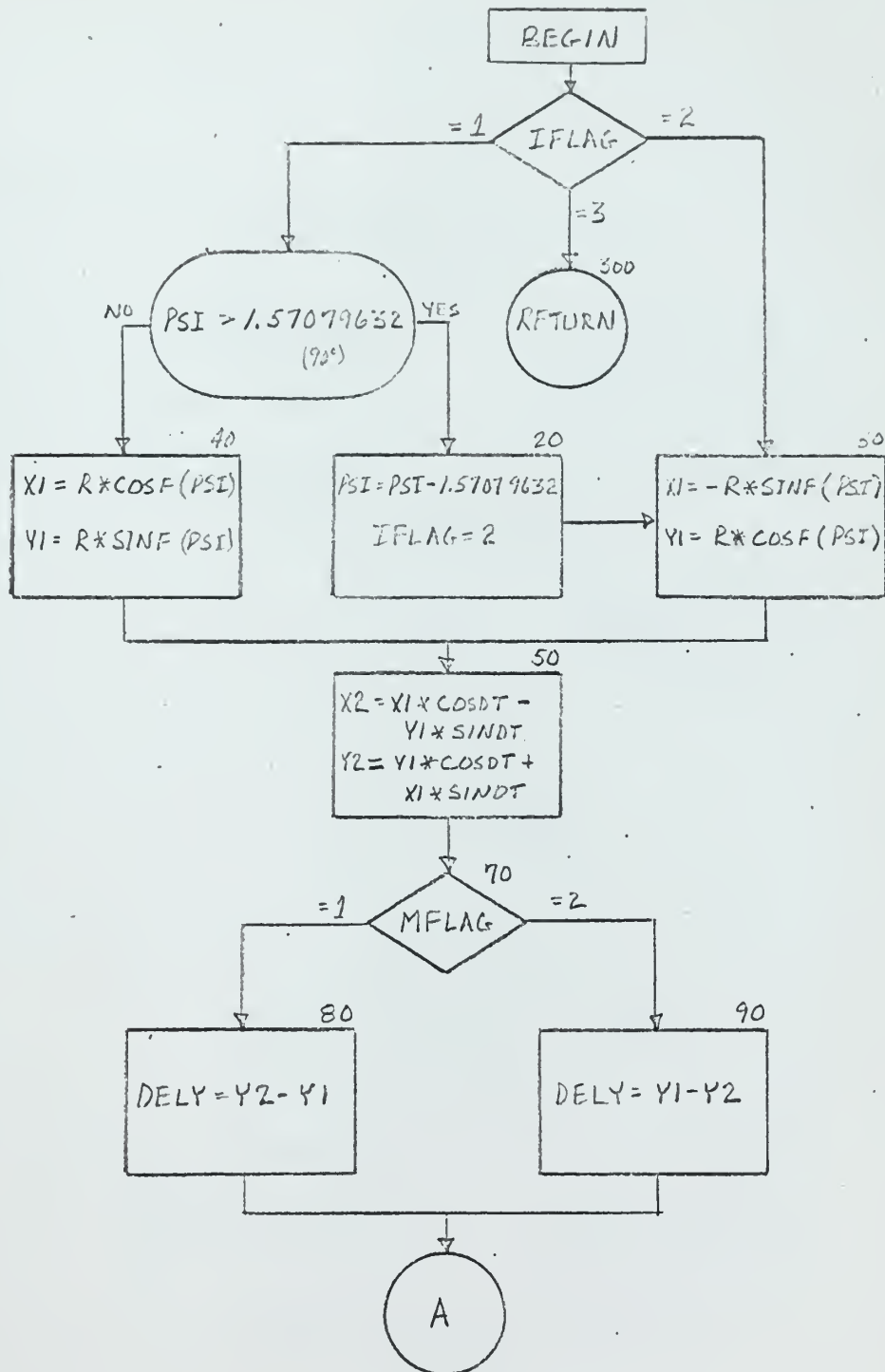




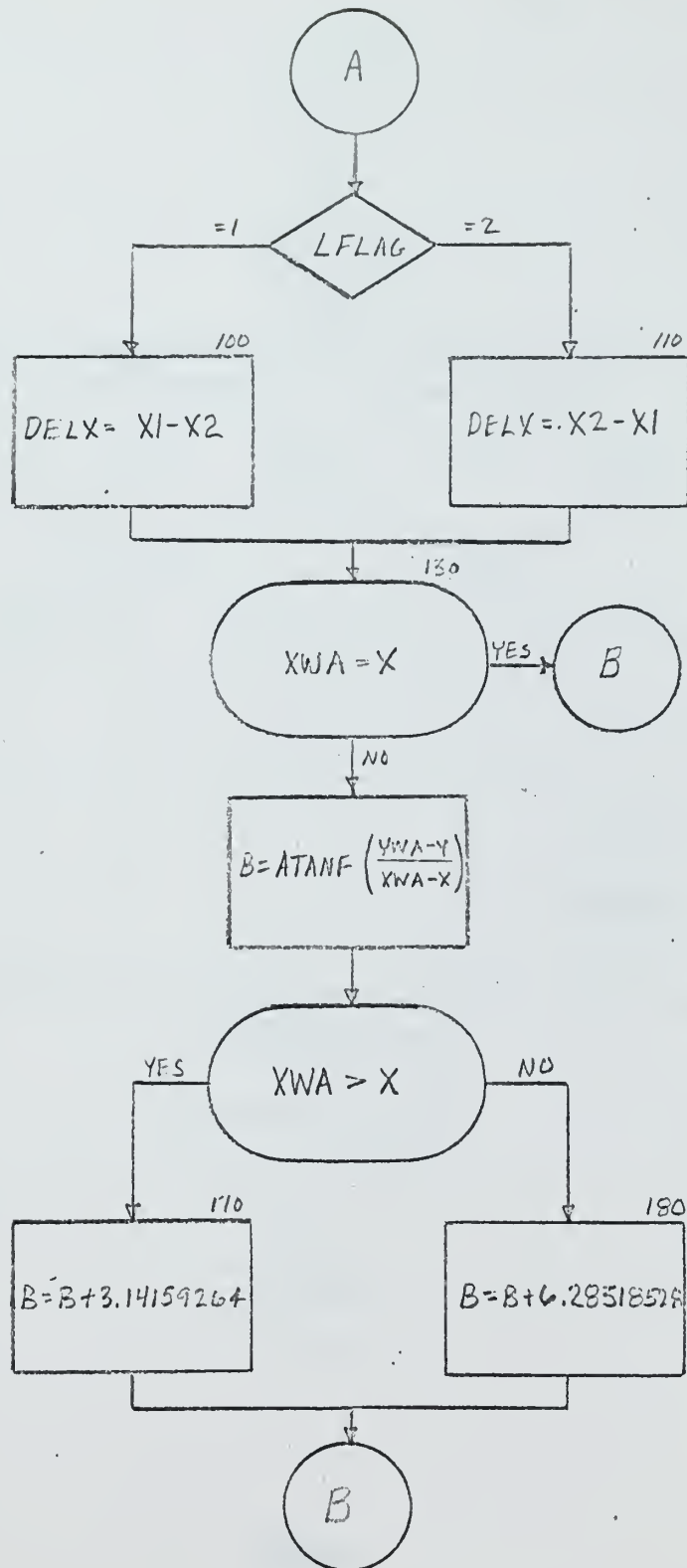
9.0

SUBROUTINE
TURN

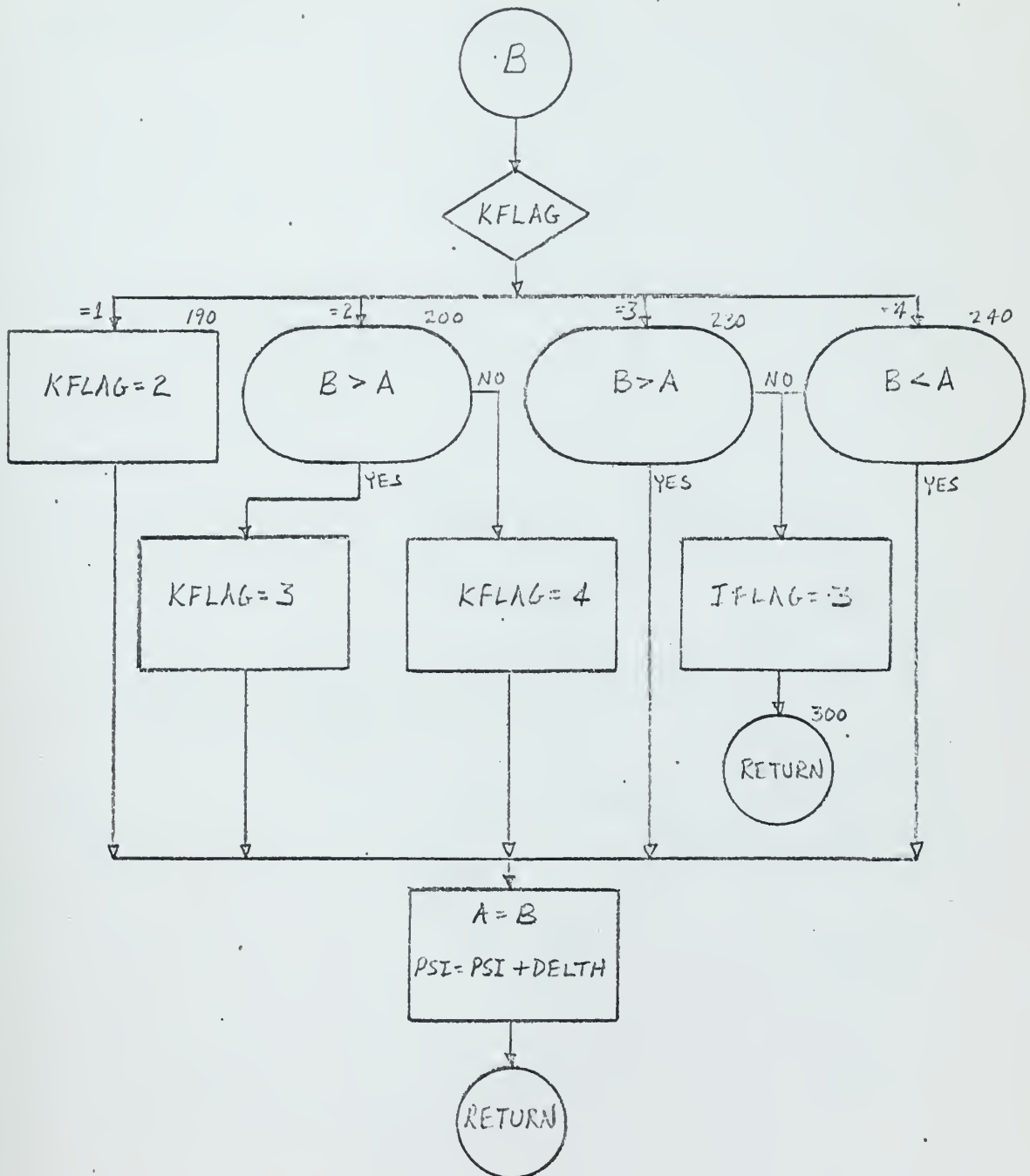
FLOWCHART 7.0
PAGE 1 OF 3







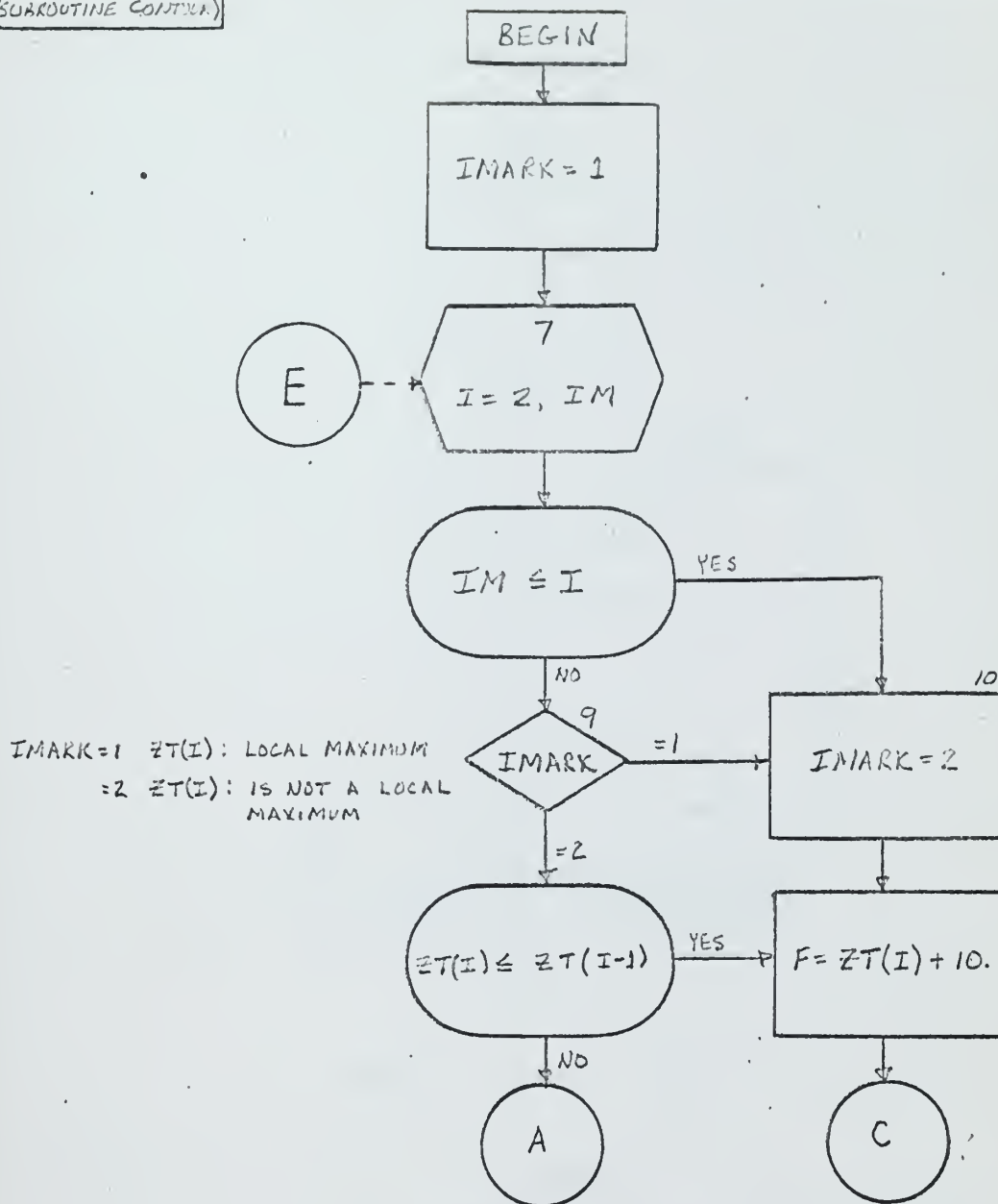




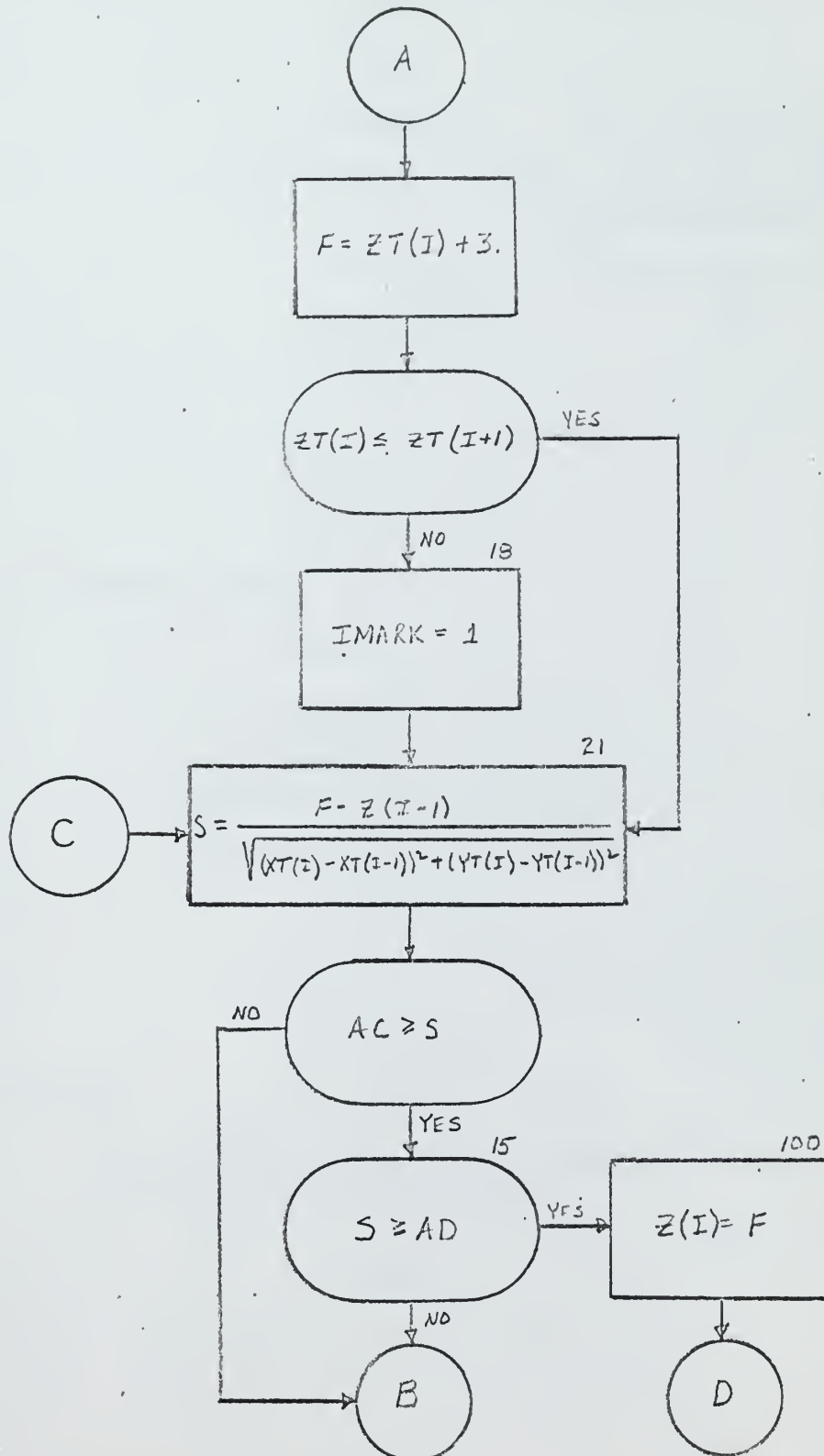
10.0

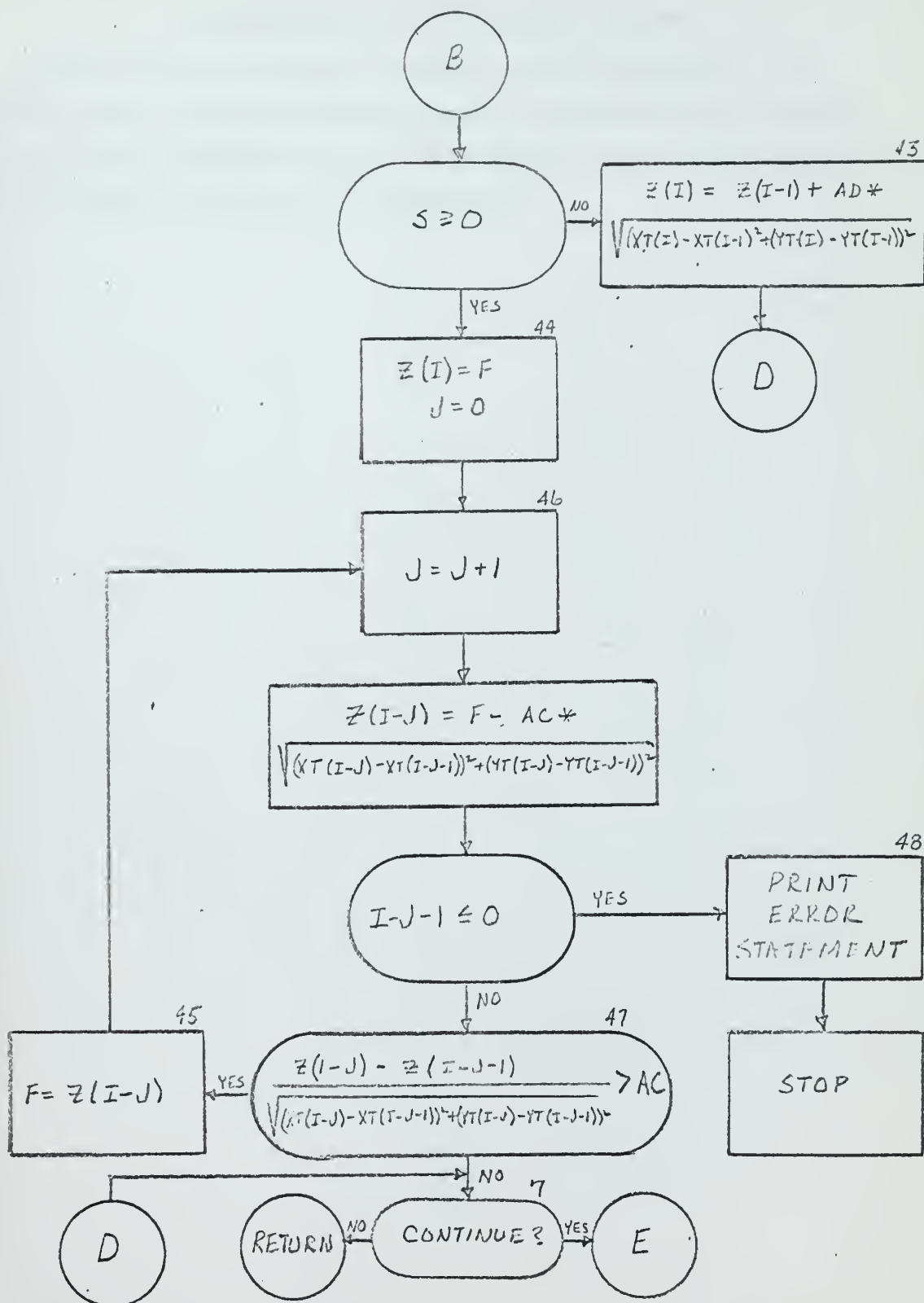
COMPUTE
NOE
FLIGHTPATH
(SUBROUTINE CONTINUED)

FLOW CHART 10.0
PAGE 1 OF 3









APPENDIX II

FORTRAN Listing of the PENAIR Program

PENAIR consists of the main program and four subroutines. The variable names used in the PENAIR program are defined in the comments appearing in the FORTRAN listing. The variable names of input and output parameters are explained in Appendix III.


```

PEN0001  PROGRAM PENAIR
PEN0002  COMMON/RANG1/XW(10,20),YW(10,20),ZW(10,20),IMAX,JMAX(10),
PEN0003  TIV(10,20),RMAX(10),TA(10),RLT(10),FT(10),RMIN(10),
PEN0004  X,Y,Z,XWA,XWB,YWA,YWB
PEN0005  1 /RANG2/ITIV(10,20),PS(10,20),JMARK,
PEN0006  P(10,20),FTA(10)
PEN0007  2 /ELEV1/PHI(12,25,25),MMAX(12),NMAX(12),BCX1,BCX2,BCX3,BCX4,
PEN0008  A BCY1,BCY2,BCY3,XA,YA,ZA,BCX,BCY
PEN0009  3 /TURN1/DELTH,R,DELX,DELY,DELD,PSI,IFLAG,KFLAG,LFLAG,MFLAG,
PEN0010  A COSDT,SINDT
PEN0011  4 /CONT1/AC,AD,IM,XT(300),YT(300),ZT(300),ZH(300)
PEN0012  DIMENSION WPNA(10),WPNB(10),NW(10,20),IK(100),AFIELD(16)
PEN0013  TYPE INTEGER TIV,TA,RLT,FT,FTA
PEN0014
PEN0015  C SLOPE(A,B,C)=(A*B)/(60.*C) $ SLOPE2(A)=.00508*TI*A/DELD
PEN0016  C COMPUTES THE SLOPE OF THE MAXIMUM CLIMB AND DESCENT OF THE A/C.
PEN0017  C
PEN0018  C
PEN0019  C
PEN0020  READ 1010,NRUN
PEN0021  READ 1010,KMAX,XMIN,YMIN,XMAX,YMAX,TI,TPD
PEN0022  1010 FORMAT(I10,6F10.0)
PEN0023  C KMAX=NUMBER OF POLYNOMIALS TO APPROXIMATE THE TERRAIN.
PEN0024  C
PEN0025  C XMIN=X COORDINATE OF LEFT EDGE OF THE TERRAIN.
PEN0026  C YMIN=Y COORDINATE OF BOTTOM EDGE
PEN0027  C XMAX=X COORDINATE OF RIGHT EDGE
PEN0028  C YMAX=Y COORDINATE OF TOP EDGE
PEN0029  C TI=GAME TIME INTERVAL (SECONDS)
PEN0030  C TPD=DIMENSION OF TERRAIN SQUARE APPROXIMATED BY ONE POLYNOMIAL.
PEN0031  C
PEN0032  IF(KMAX.GT.12)9,10
PEN0033  9 PRINT 1008
PEN0034  1008 FORMAT(1H1////77H NUMBER OF APPROXIMATING TERRAIN POLYNOMIALS EXCE
PEN0035  *EDS 12. THIS RUN TERMINATED. )
PEN0036  STOP
10 READ 1090,X,Y,Z,X2,Y2
1090 FORMAT(5F10.0)

```



```

C      X,Y,Z ARE THE BEGINNING COORDINATES OF THE AIRCRAFT.
C      X2,Y2 ARE THE COORDINATES OF THE DESIRED DESTINATION OF THE A/C.
C
C      20 READ 1090,GMAX,RC,RD,GS
C      GMAX=MAXIMUM POSITIVE G TO BE USED BY THE AIRCRAFT.
C      RC=MAXIMUM RATE OF CLIMB (FT./MIN)
C      RD=MAXIMUM RATE OF DESCENT (FT./MIN)
C      GS=ESTIMATED GROUND SPEED (KNOTS)
C
C      READ 1010,IMAX
C      IMAX=THE NUMBER OF TYPES OF WEAPONS
C
C      IF(IMAX.GT.10)15,16
C          15 PRINT 1015
C          1015 FORMAT(1H1////56H WEAPON DIMENSIONS ARE EXCEEDED. THIS RUN IS TERMPEN0037
C              *INATED. )
C              STOP
C          16 DO 40 I=1,IMAX
C
C              READ 1030, WPNA(I),WPNB(I),JMAX(I),TA(I),FT(I),RLT(I),RMAX(I),
C                  1RMIN(I)
C              1030 FORMAT(2A8,4X,4I5,2F10.0)
C              WPNA,WPNB=TYPE OF WEAPON (COL.1-16)
C              JMAX(I)=NUMBER OF WEAPONS OF TYPE I
C              TA=TIME REQUIRED TO ACQUIRE TARGET (SEC.)
C              FT=FIRING TIME BEFORE RELOADING (SEC.)
C              RLT=TIME REQUIRED TO RELOAD (SEC.)
C              RMAX=MAXIMUM RANGE OF WEAPON (METERS)
C              RMIN=MINIMUM RANGE OF WEAPON (METERS)
C
C              IF(JMAX(I).GT.20)15,17
C              17 READ 1040, (P(I,K),K=1,20)
C              1040 FORMAT(10F8.0)
C              P(I,K)=PROBABILITY OF A KILL BY WEAPON I AT SLANT RANGE K
C
C              JMAXX=JMAX(I)

```



```

C      40 READ 1050,(NW(I,J),XW(I,J),YW(I,J),ZW(I,J),J=1,JMAXX)
C      1050 FORMAT(I2,8X,3F10.0)
C      NW=NUMBER ASSIGNED TO THE WEAPON
C      XW,YW=X AND Y COORDINATES OF THE WEAPON
C
C      DO 96 I=1,KMAX
C
C      READ 1060, MMAX(I),NMAX(I)
C      1060 FORMAT(2I10)
C      MMAX=DEGREE OF X IN THE ITH TERRAIN POLYNOMIAL
C      NMAX=DEGREE OF Y
C      MMAXX=MMAX(I)+1 $ NMAXX=NMAX(I)+1
C
C      IF(MMAX(I).GT.25.OR.NMAX(I).GT.25)95,96
C      95 PRINT 1095
C      1095 FORMAT(1H1////78H MAXIMUM DEGREE OF THE TERRAIN POLYNOMIAL IS EXCEEDING
C      *EDED. THIS RUN IS TERMINATED. )
C      STOP
C      96 READ 1070,((PHI(I,K,L),L=1,NMAXX),K=1,MMAXX)
C      1070 FORMAT(4E20.10)
C      PHI(I,K,L)=COEFFICIENTS OF TERRAIN POLYNOMIAL
C      I=ITH TERRAIN POLYNOMIAL
C      K=DEGREE OF X
C      L=DEGREE OF Y
C
C      BCX=XMIN-TPD $ BCY=YMIN-TPD
C      BCX1 = XMIN+TPD/2. $ BCY1 = YMIN+TPD/2.
C      BCX2=BCX1+TPD $ BCX3=BCX2+TPD $ BCX4=BCX3+TPD
C      BCY2=BCY1+TPD. $ BCY3=BCY2+TPD
C      PRINT 2025
C      2025 FORMAT (1H1///12X,22H SUMMARY OF WEAPON DATA //)
C      PRINT 2020,(WPNA(I),WPNB(I),I=1,IMAX)
C      2020 FORMAT(1H0, 11HWEAPON TYPE, 10X, 12A8)
C      PRINT 2024,(JMAX(I),I=1,IMAX)
C      2024 FORMAT( 1X, 17HNUMBER OF WEAPONS, 110,5I16)

```

PEN0073
 PEN0074
 PEN0075
 PEN0076
 PEN0077
 PEN0078
 PEN0079
 PEN0080
 PEN0081
 PEN0082
 PEN0083
 PEN0084
 PEN0085
 PEN0086
 PEN0087
 PEN0088
 PEN0089
 PEN0090
 PEN0091
 PEN0092
 PEN0093
 PEN0094
 PEN0095
 PEN0096
 PEN0097
 PEN0098
 PEN0099
 PEN0100
 PEN0101
 PEN0102
 PEN0103
 PEN0104
 PEN0105
 PEN0106
 PEN0107
 PEN0108


```

PRINT 2026,(TA(I),I=1,IMAX)
2026 FORMAT( 1X, 22HACQUISITION TIME (SEC), I5,5I16)
PRINT 2028,(FT(I),I=1,IMAX)
2028 FORMAT( 1X, 17HFIRING TIME (SEC), I10,5I16)
PRINT 2030,(RLT(I),I=1,IMAX)
2030 FORMAT( 1X, 20HRELOADING TIME (SEC),I7,5I16)
PRINT 2032,(RMIN(I),I=1,IMAX)
2032 FORMAT( 1X, 22HMINIMUM RANGE (METERS), F6.0,5F16.0)
PRINT 2034,(RMAX(I),I=1,IMAX)
2034 FORMAT( 1X, 22HMAXIMUM RANGE (METERS),F6.0,5F16.0)
PRINT 2036
2036 FORMAT(1H0, 6X, 6HWEAPON, 9X, 4HSITE, 6X, 6HWEAPON /
      8X 4HTYPE, 9X, 6HNUMBER, 5X, 9HPOSITIONS /)
DO 180 I=1,IMAX
C
C   CONVERT KILL PROBABILITIES TO SURVIVAL PROBABILITIES.
DO 30 K=1,20
30 P(I,K)=1.-P(I,K)
C
      TA(I)=TA(I)/TI
      FT(I)=FT(I)/TI
      FTA(I)=TA(I)+FT(I)
      RLT(I)=RLT(I)/TI
      RMAX(I)=RMAX(I)**2 $ RMIN(I)=RMIN(I)**2
      PRINT 2038,WPNA(I),WPNB(I)
2038 FORMAT( 5X,2A8)
      JMAXX=JMAX(I)
      DO 180 J=1,JMAXX $ ICJ=I+10*(J-1)
      XA=XW(ICJ) $ YA=YW(ICJ) $ CALL ELEV
C
C   ASSIGN THE HIGHER OF THE ACTUAL OR COMPUTED ELEVATION TO THE WPN.
      IF(ZA.GT.ZW(ICJ))50,180
      50 ZW(ICJ) = ZA
C
      180 PRINT 2040, NW(ICJ),XW(ICJ),YW(ICJ),ZW(ICJ)
2040 FORMAT(22X,I3,5X,2F7.0,F5.0)

```

PEN0109
 PEN0110
 PEN0111
 PEN0112
 PEN0113
 PEN0114
 PEN0115
 PEN0116
 PEN0117
 PEN0118
 PEN0119
 PEN0120
 PEN0121
 PEN0122
 PEN0123
 PEN0124
 PEN0125
 PEN0126
 PEN0127
 PEN0128
 PEN0129
 PEN0130
 PEN0131
 PEN0132
 PEN0133
 PEN0134
 PEN0135
 PEN0136
 PEN0137
 PEN0138
 PEN0139
 PEN0140
 PEN0141
 PEN0142
 PEN0143
 PEN0144


```

2041 PRINT 2041
      FORMAT (1H1)
C
C   COMPUTE THE MAXIMUM ANGLE OF BANK OF THE A/C.
      AB=ACOSF(1./GMAX)
C
      XX=X $ YY=Y $ ZZ=Z
      DO 9990 N=1,NRUN
      READ 1000,(AFIELD(I),I=1,16)
      FORMAT(4A8)
      THIS READS IN THE TITLE OF THE RUN
C
      READ 1075,IMARK,IPUNCH,GSNEW
      FORMAT(2I10,F10.0)
      IMARK=1 FLIGHTPATH UPON RECEIVING FIRE STRAIGHT AND LEVEL
      2 CLIMB STRAIGHT AHEAD
      3 TURN AWAY, LEVEL
      4 TURN AWAY AND CLIMB
      5 DIVE AND FLY NOE
      6 TURN AWAY AND FLY NOE
      7 COMPLETELY PREPLANNED
      8 X AND Y COORDINATES PREPLANNED, NOE COMPUTED
      IPUNCH=1 A/C FLIGHT PATH WILL BE PUNCHED ON CARDS.
      =0 A/C FLIGHT PATH WILL NOT BE PUNCHED ON CARDS.
      GSNEW=GROUND SPEED AFTER RECEIVING FIRE (KNOTS)
      IF(IPUNCH)182,183,182
      182 PUNCH 1001,(AFIELD(I),I=1,4)
      1001 FORMAT (4A8)
      183 PRINT 2000,(AFIELD(I),I=1,4)
      2000 FORMAT (///31X,10HSUMMARY OF/28X,4A8)
      PRINT 2010,(AFIELD(I),I=5,16)
      2010 FORMAT (19X,10HMISSION - ,4A8/(29X,4A8))
      DO 181 I=1,IMAX $ JMAXX=JMAXX(I)
      DO 181 J=1,JMAXX
      ITIV(I,J)=TIV(I,J)=0
      181 PS(I,J)=1.

```

PEN0145
 PEN0146
 PEN0147
 PEN0148
 PEN0149
 PEN0150
 PEN0151
 PEN0152
 PEN0153
 PEN0154
 PEN0155
 PEN0156
 PEN0157
 PEN0158
 PEN0159
 PEN0160
 PEN0161
 PEN0162
 PEN0163
 PEN0164
 PEN0165
 PEN0166
 PEN0167
 PEN0168
 PEN0169
 PEN0170
 PEN0171
 PEN0172
 PEN0173
 PEN0174
 PEN0175
 PEN0176
 PEN0177
 PEN0178
 PEN0179
 PEN0180


```

JMARK=KMARK=LMARK=NMARK=IFLAG=KFLAG=1 $ DELZ=0.
X=XX $ Y=YY $ Z=ZZ
IF(IMARK.LE.6) 220,100
C
C COMPUTE THE DISTANCE (METERS) TRAVELED BY THE A/C IN ONE TIME
C INTERVAL.
220 DELD=0.5144444444*TI*GS
C
C COMPUTE THE SLOPE OF THE MAXIMUM CLIMB AND DESCENT OF THE A/C.
AC=SLOPE(RC,TI,GS) $ AD=-SLOPE(RD,TI,GS)
C COMPUTE THE SPEED TO BE FLOWN UPON RECEIVING FIRE.
DELDNEW=0.5144444444*TI*GSNEW $ DELDT=DELDNEW-DELD $ GO TO 281
C
100 I=0
110 I=I+1
C
IF(I.GT.300)111,112
111 PRINT 1111
1111 FORMAT(1H1///72H PREPLANNED FLIGHTPATH INPUT POINTS EXCEEDS 300.
*THIS RUN IS TERMINATED. )
STOP
112 READ 1080,NPATH,XT(I),YT(I),ZH(I)
1080 FORMAT(5X,I2,10X,3F9.1)
C NPATH=PREPLANNED FLIGHT PATH NUMBER (ANY POSITIVE INTEGER )
C XG(I)=PREPLANNED X COORDINATE IN THE ITH TIME INTERVAL
C YG(I)=PREPLANNED Y COORDINATE
C ZH(I)=PREPLANNED Z COORDINATE
C
IF(NPATH)110,120,110
120 IM=I-1
DO 130 I=1,IM
XA=XT(I) $ YA=YT(I) $ CALL ELEV
130 ZT(I)=ZA
IMARK=IMARK-6 $ GO TO (540,260)IMARK
260 AC=SLOPE(RC,TI,GS) $ AD=-SLOPE(RD,TI,GS) $ GO TO 530
230 IF(DELDT)210,234,213

```



```

210 IF(DELD.LE.DEKDNEW)214,215
214 DELDT=0. $ GO TO 234
213 IF(DELD.GE.DEKDNEW)214,216
215 DELD=DELD-1.544444444*TI $ GO TO 280
216 DELD=DELD+1.028888888*TI
280 GO TO(281,281,298,298,281,298)IMARK
281 IF(X.EQ.X2)282,284
282 DELX=0. $ GO TO 288
284 DELX = DELD/SQRTF(1.+((Y2-Y)/(X2-X))**2)
    IF(X2.GT.X)288,291
291 DELX = -DELX
288 IF(Y.EQ.Y2)286,292
286 DELY=0.
    GO TO 299
292 DELY = DELD/SQRTF(1.+((X2-X)/(Y2-Y))**2)
    IF(Y2.GT.Y)299,293
293 DELY =--DELY
299 PSI=ABSF(ATANF(DELDX/DELY))
298 AC=SLOPE2(RC) $ AD=--SLOPE2(RD)
C
C
C
    COMPUTE THE RADIUS OF TURN OF THE A/C (METERS).
    R=(DELD**2)/(TI**2*9.80616*TANF(AB)) $ DELTH=DELD/R
C
    COSDT=COSF(DELDX) $ SINDT=SINF(DELDX)
234 X=X+DELD $ Y=Y+DELY $ Z=Z+DELZ
    IF(X.LT.XMIN)238,231
231 IF(X.GT.XMAX)238,232
232 IF(Y.LT.YMIN)238,233
233 IF(Y.GT.YMAX)238,235
238 GO TO (600,520)LMARK
235 IF(IPUNCH)237,236,237
237 XA=X $ YA=Y $ CALL ELEV
    PUNCH 4000,X,Y,Z,ZA
4000 FORMAT (6X,1H1,10X,4F9.1)
236 GO TO (239,370,510)LMARK
239 CALL RANGE
PEN0217
PEN0218
PEN0219
PEN0220
PEN0221
PEN0222
PEN0223
PEN0224
PEN0225
PEN0226
PEN0227
PEN0228
PEN0229
PEN0230
PEN0231
PEN0232
PEN0233
PEN0234
PEN0235
PEN0236
PEN0237
PEN0238
PEN0239
PEN0240
PEN0241
PEN0242
PEN0243
PEN0244
PEN0245
PEN0246
PEN0247
PEN0248
PEN0249
PEN0250
PEN0251
PEN0252

```



```

240 GO TO (234,245) JMARK
C
C      JMARK=1 IF NO WEAPONS HAVE FIRED.
C      =2 IF ANY WEAPON HAS FIRED.
C      IF JMARK=2, THEN
C      LMARK=1 IF IMARK=1,2,3,4
C      =2 IF      =6
C      =3 IF      =5
C
245 GO TO(230,250,301,250,500,302)IMARK
250 DELZ=AC*DELD $ IMARK=IMARK-1 $ GO TO 230
302 LMARK=2
301 GO TO (303,370)NMARK
C
C      NMARK=1 BEFORE TURN DIRECTION HAS BEEN DETERMINED.
C      =2 AFTER  TURN DIRECTION HAS BEEN DETERMINED.
C
303 XWA=XWB $ YWA=YWB
    IF(DELDX)305,360,310
305 LFLAG=2 $ GO TO 315
310 LFLAG=1
315 IF(DELY)325,320,330
325 MFLAG=2 $ GO TO 340
330 MFLAG=1 $ GO TO 340
320 PRINT 7000
7000 FORMAT(1H1,10X,53HILLEGAL INPUT, DELY EQUALS ZERO, THIS RUN TERMINATED.)
    STOP
360 IF(X.GT.XWA)350,355
350 LFLAG=1 $ MFLAG=1 $ GO TO 340
355 LFLAG=2 $ MFLAG=1
340 II=0 $ NMARK=2
370 CALL TURN
    GO TO (230,510)LMARK
500 II=0 $ LMARK=3
510 II=II+1 $ XT(II)=XA=X $ YT(II)=YA=Y

```

```

PEN0253
PEN0254
PEN0255
PEN0256
PEN0257
PEN0258
PEN0259
PEN0260
PEN0261
PEN0262
PEN0263
PEN0264
PEN0265
PEN0266
PEN0267
PEN0268
PEN0269
PEN0270
PEN0271
PEN0272
PEN0273
PEN0274
PEN0275
PEN0276
PEN0277
PEN0278
PEN0279
PEN0280
PEN0281
PEN0282
PEN0283
PEN0284
PEN0285
PEN0286
PEN0287
PEN0288

```



```

CALL ELEV $ ZT(II)=ZA
GO TO 230
520 IM=II $ ZH(1)=Z
PUNCH 3030
3030 FORMAT(7HCONTOUR)
530 CALL CONTOUR
540 DO 550 I=1,IM
X=XT(I) $ Y=YT(I) $ Z=ZH(I)
IF(IPUNCH)545,550,545
545 PUNCH 4000,X,Y,Z,ZT(I)
550 CALL RANGE
C
C COMPUTE THE SURVIVAL PROBABILITY OF THE A/C.
600 TPS=1.
DO 802 I=1,IMAX $ JMAXX=JMAX(I)
DO 802 J=1,JMAXX
802 TPS=TPS*PS(I,J)
PRINT 2050
2050 FORMAT(//// 8X,65HSUMMARY OF TIME IN VIEW AND SURVIVAL PROBABILITY BY WEAPON NUMBER )
PRINT 2060
2060 FORMAT(1H0,13HWEAPON NUMBER)
778 DO 780 I=1,IMAX
JMAXX=JMAX(I)
780 PRINT 2070,(NW(I,J),J=1,JMAXX)
2070 FORMAT(21X,10I8)
PRINT 2080
2080 FORMAT(1H0,12HTIME IN VIEW)
DO 781 I=1,IMAX
JMAXX=JMAX(I)
781 PRINT 2070,(ITIV(I,J),J=1,JMAXX)
PRINT 2090
2090 FORMAT(1H0,20HSURVIVAL PROBABILITY)
DO 801 I=1,IMAX $ JMAXX=JMAX(I)
801 PRINT 3000,(PS(I,J),J=1,JMAXX)
3000 FORMAT(21X,10F8.4)
PEN0289
PEN0290
PEN0291
PEN0292
PEN0293
PEN0294
PEN0295
PEN0296
PEN0297
PEN0298
PEN0299
PEN0300
PEN0301
PEN0302
PEN0303
PEN0304
PEN0305
PEN0306
PEN0307
PEN0308
PEN0309
PEN0310
PEN0311
PEN0312
PEN0313
PEN0314
PEN0315
PEN0316
PEN0317
PEN0318
PEN0319
PEN0320
PEN0321
PEN0322
PEN0323
PEN0324

```



```

PEN0325
PEN0326
PEN0327
PEN0328
PEN0329
PEN0330
PEN0331
PEN0332
PEN0333
PEN0334
PEN0335
PEN0336
PEN0337
PEN0338
PEN0339
PEN0340
PEN0341
PEN0342
PEN0343
PEN0344
PEN0345
PEN0346
PEN0347
PEN0348
PEN0349
PEN0350
PEN0351
PEN0352
PEN0353
PEN0354
PEN0355
PEN0356
PEN0357
PEN0358
PEN0359
PEN0360

3010 PRINT 3010,TPS
      FORMAT(//30X,26HTOTAL SURVIVAL PROBABILITY ,F6.4/1H1)
      END FILE 52
3025 WRITE (1,3025)
      FORMAT(19H PRESS START SWITCH )
      PAUSE 1
9990 CONTINUE
9992 STOP
      END

SUBROUTINE RANGE
C
C THIS SUBROUTINE DETERMINES THE SURVIVAL PROBABILITY OF THE A/C.
C
COMMON/RANG1/XW(10,20),YW(10,20),ZW(10,20),IMAX,JMAX(10),
A TIV(10,20),RMAX(10),TA(10),RLT(10),FT(10),RMIN(10),
B X,Y,Z,XWA,XWB,YWA,YWB
1 /RANG2/ITIV(10,20),PS(10,20),JMARK,
A P(10,20),FTA(10)
2 /ELEV1/PHI(12,25,25),MMAX(12),BCX1,BCX2,BCX3,BCX4,
A BCY1,BCY2,BCY3,XL,YL,ZT,BCX,BCY
A TYPE INTEGER TIV,TA,RLT,FTA
DO 10 I=1,IMAX $ JMAXX=JMAX(I)
DO 10 J=1,JMAXX
ICJ=I+10*(J-1)
XA=(X-XW(ICJ))*2 $ YA=(Y-YW(ICJ))*2 $ ZA=(Z-ZW(ICJ))*2
C
C DETERMINES IF THE TERRAIN INTERCEPTS THE LINE OF SIGHT.
RW=10. $ TR=SQRTF(XA+YA) $ TAN=(Z-ZW(ICJ))/TR
SIN=(XW(ICJ)-X)/TR $ COS=(YW(ICJ)-Y)/TR
20 ZL=RW*TAN+ZW(ICJ) $ XL=XW(ICJ)-RW*SIN $ YL=YW(ICJ)-RW*COS
CALL ELEV
IF(ZT.GT.ZL)70,40

```



```

C      70 TIV(ICJ)=0 $ GO TO 10
      40 RW=RW+50.
        IF(RW.GT.TR)50,20
      50 TIV(ICJ)=TIV(ICJ)+1
        ITIV(ICJ)=ITIV(ICJ)+1
        IF(TIV(ICJ).LE.TA(I))10,80
      80 RP=XA+YA+ZA
        IF(RP.GT.RMAX(I))83,86
      83 TIV(ICJ)=TIV(ICJ)-1 $ GO TO 10
      86 IF(RP.GT.RMIN(I))81,83
      81 IF(TIV(ICJ).GT.FTA(I))82,84
      82 TIV(ICJ)=TA(I)/2-RLT(I) $ GO TO 10
C      THE APPROPRIATE SURVIVAL PROBABILITY IS SELECTED.
C
C      84 K=SQRTF(RP)/101
      99 PS(ICJ)=PS(ICJ)*P(I,K)
        JMARK=2 $ XWB=XW(ICJ) $ YWB=YW(ICJ)
      10 CONTINUE
      60 RETURN
      END
C
C      SUBROUTINE ELEV
C      THIS SUBROUTINE DETERMINES THE TERRAIN ELEVATION, GIVEN X AND Y.
C      COMMON/ELEV1/PHI(12,25,25),MMAX(12),NMAX(12),BCX1,BCX2,BCX3,BCX4,
A      BCY1,BCY2,BCY3,XT,YT,ZT,BCX,BCY
C      DETERMINE WHICH POLYNOMIAL TO SOLVE.
C      INX=(XT-BCX)/1000. $ INY=(YT-BCY)/1000.
C
PEN0361
PEN0362
PEN0363
PEN0364
PEN0365
PEN0366
PEN0367
PEN0368
PEN0369
PEN0370
PEN0371
PEN0372
PEN0373
PEN0374
PEN0375
PEN0376
PEN0377
PEN0378
PEN0379
PEN0380
PEN0381
PEN0382
PEN0383
PEN0384
PEN0385
PEN0386
PEN0387
PEN0388
PEN0389
PEN0390
PEN0391
PEN0392
PEN0393
PEN0394
PEN0395
PEN0396

```



```

PEN0397
PEN0398
PEN0399
PEN0400
PEN0401
PEN0402
PEN0403
PEN0404
PEN0405
PEN0406
PEN0407
PEN0408
PEN0409
PEN0410
PEN0411
PEN0412
PEN0413
PEN0414
PEN0415
PEN0416
PEN0417
PEN0418
PEN0419
PEN0420
PEN0421
PEN0422
PEN0423
PEN0424
PEN0425
PEN0426
PEN0427
PEN0428
PEN0429
PEN0430
PEN0431
PEN0432

I=INX+4*(INY-1)
GO TO (1,2,3,4,5,6,7,8,9,10,11,12)I
1 X=XT-BCX1 $ Y=YT-BCY1 $ GO TO 30
2 X=XT-BCX2 $ Y=YT-BCY1 $ GO TO 30
3 X=XT-BCX3 $ Y=YT-BCY1 $ GO TO 30
4 X=XT-BCX4 $ Y=YT-BCY1 $ GO TO 30
5 X=XT-BCX1 $ Y=YT-BCY2 $ GO TO 30
6 X=XT-BCX2 $ Y=YT-BCY2 $ GO TO 30
7 X=XT-BCX3 $ Y=YT-BCY2 $ GO TO 30
8 X=XT-BCX4 $ Y=YT-BCY2 $ GO TO 30
9 X=XT-BCX1 $ Y=YT-BCY3 $ GO TO 30
10 X=XT-BCX2 $ Y=YT-BCY3 $ GO TO 30
11 X=XT-BCX3 $ Y=YT-BCY3 $ GO TO 30
12 X=XT-BCX4 $ Y=YT-BCY3

C
C
C
SOLVE THE POLYNOMIAL, ZCOMP =.F(X,Y)

30 ZCOMP=0.
ICJ=I+12*(MMAX(I)-1)
40 ICJCK=ICJ+300*(NMAX(I)-1) $ POLY=PHI(ICJCK)
50 ICJCK=ICJCK-300 $ POLY=POLY*Y+PHI(ICJCK)
IF(ICJCK-ICJ)100,70,50
70 ZCOMP=ZCOMP*X+POLY $ IF(ICJ-I)100,90,80
80 ICJ=ICJ-12 $ GO TO 40

C
C
C
CORRECT FEET TO METERS

90 ZT=0.304801*ZCOMP $ RETURN
100 PRINT 1000
1000 FORMAT(1H1,5X,31HERROR IN ELEVATION COMPUTATION. )
STOP
END

```



```

C
C
C
SUBROUTINE TURN
THIS SUBROUTINE TURNS THE AIRCRAFT AWAY FROM A FIRING WEAPON.

COMMON/RANG1/XW(10,20),YW(10,20),ZW(10,20),IMAX,JMAX(10),
A      TIV(10,20),RMAX(10),TA(10),RLT(10),FT(10),RMIN(10),
B      X,Y,Z,XWA,XWB,YWA,YWB
3      /TURN1/DELTH,R,DELX,DELY,DELD,PSI,IFLAG,KFLAG,LFLAG,MFLAG,
A      COSDT,SINDT
      GO TO (10,30,300)IFLAG
10 IF(PSI.GT.1.57079632)20,40
20 PSI=PSI-1.57079632 $ IFLAG=2
30 X1=-R*SINF(PSI) $ Y1=R*COSF(PSI) $ GO TO 50
40 X1=R*COSF(PSI) $ Y1=R*SINF(PSI)
50 X2=X1*COSDT-Y1*SINDT $ Y2=Y1*COSDT+X1*SINDT
70 GO TO (80,90) MFLAG
80 DELY=Y2-Y1 $ GO TO (100,110)LFLAG
90 DELY=Y1-Y2 $ GO TO (100,110) LFLAG
100 DELX=X1-X2 $ GO TO 130
110 DELX=X2-X1
130 IF(XWA.EQ.X) 140,150
140 B=0.78539816 $ GO TO (190,200,230,240) KFLAG
150 B=ATANF((YWA-Y)/(XWA-X))
160 IF(XWA.GT.X) 170,180
170 B=B+3.14159264 $ GO TO (190,200,230,240) KFLAG
180 B=B+6.28318528 $ GO TO (190,200,230,240) KFLAG
190 KFLAG=2 $ GO TO 250
200 IF(B.GT.A) 210,220
210 KFLAG=3 $ GO TO 250
220 KFLAG=4 $ GO TO 250
230 IF(B.GT.A) 250,260
240 IF(B.LT.A) 250,260
250 A=B $ PSI=PSI+DELTH $ RETURN
260 IFLAG=3
300 RETURN
      END
PEN0433
PEN0434
PEN0435
PEN0436
PEN0437
PEN0438
PEN0439
PEN0440
PEN0441
PEN0442
PEN0443
PEN0444
PEN0445
PEN0446
PEN0447
PEN0448
PEN0449
PEN0450
PEN0451
PEN0452
PEN0453
PEN0454
PEN0455
PEN0456
PEN0457
PEN0458
PEN0459
PEN0460
PEN0461
PEN0462
PEN0463
PEN0464
PEN0465
PEN0466
PEN0467
PEN0468

```


PEN0469
 PEN0470
 PEN0471
 PEN0472
 PEN0473
 PEN0474
 PEN0475
 PEN0476
 PEN0477
 PEN0478
 PEN0479
 PEN0480
 PEN0481
 PEN0482
 PEN0483
 PEN0484
 PEN0485
 PEN0486
 PEN0487
 PEN0488
 PEN0489
 PEN0490
 PEN0491
 PEN0492
 PEN0493
 PEN0494
 PEN0495
 PEN0496
 PEN0497
 PEN0498
 PEN0499
 PEN0500
 PEN0501
 PEN0502
 PEN0503
 PEN0504

SUBROUTINE CONTOUR

THIS SUBROUTINE COMPUTES A NAP OF THE EARTH FLIGHT PATH.

COMMON/CONT1/AC,AD,IM,XT(300),YT(300),ZT(300),Z(300)

IMARK=1

DO 7 I=2,IM

IF(IM.LE.I)10,9

9 GO TO (10,11)IMARK

10 IMARK=2

GO TO 19

11 IF(ZT(I).LE.ZT(I-1))19,17

17 F=ZT(I)+3.

IF(ZT(I).LE.ZT(I+1))20,18

20 IMARK=2 \$ GO TO 21

18 IMARK=1 \$ GO TO 21

19 F=ZT(I)+10.

21 S=(F-Z(I-1))/SQRTF((XT(I)-XT(I-1))**2+(YT(I)-YT(I-1))**2)

IF(AC.GE.S)15,22

15 IF(S.GE.AD) 100,22

100 Z(I)=F \$ GO TO 7

22 IF(S)43,44,44

43 Z(I)=Z(I-1)+AD*SQRTF((XT(I)-XT(I-1))**2+(YT(I)-YT(I-1))**2)

GO TO 7

44 Z(I)=F

J=0

46 J=J+1

Z(I-J)=F-AC*SQRTF((XT(I-J)-XT(I-J-1))**2+(YT(I-J)-YT(I-J-1))**2)

IF(I-J-1)48,48,47

47 IF((Z(I-J)-Z(I-J-1))/SQRTF((XT(I-J)-XT(I-J-1))**2+(YT(I-J)-YT(I-J-1))**2))

*YT(I-J-1))**2)).GT.AC)45,7

45 F=Z(I-J) \$ GO TO 46



```

48 PRINT 7010
7010 FORMAT(1H1////78H AIRCRAFT BEGINNING ELEVATION IS TO LOW TO AVOID
      *TERRAIN. THIS RUN TERMINATED. )
      STOP
      7 CONTINUE
      RETURN
      END
      END
      FINIS
PEN0505
PEN0506
PEN0507
PEN0508
PEN0509
PEN0510
PEN0511
PEN0512
PEN0513

```


Appendix III

Inputs and Outputs of PENAIR

Data cards for the PENAIR program should be prepared in the following manner

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
1	1	NRUN	1-10	A right justified integer, the number of runs to be performed.
2	1	KMAX	1-10	A right justified integer, the number of squares used to approximate the terrain, $KMAX \leq 12$.
		XMIN	11-20	A right justified floating point number, the x coordinate of the left edge of the terrain. All coordinates are in meters, that is the distance between the two x coordinates 72455 and 72454 is one meter.
		YMIN	21-30	A right justified floating point number, the y coordinate of the bottom edge of the terrain.
		XMAX	31-40	The x coordinate of the right edge of the terrain.

Date Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
		YMAX	41-50	The y coordinate of the top edge of the terrain.
		TI	51-60	A right justified floating point number, the time interval of the game (seconds), $TI > 0$.
		TPD	61-70	A right justified floating point number, the dimension of the map square approximated by one polynomial (meters).
3	1	X	1-10	Right justified floating point numbers, the beginning coordinates of the aircraft. If IMARK = 3, 4, or 6, y must equal YMIN.
		Y	11-20	
		Z	21-30	
		X2	31-40	The x and y coordinates of the desired destination of the aircraft. If IMARK = 7 or 8, this card may be blank.
		Y2	41-50	
4	1	GMAX	1-10	A right justified floating point number, the maximum positive G force to be used by the aircraft.
		RC	11-20	A right justified floating point number, the aircraft's maximum rate of climb (feet/minute)

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
		RD	21-30	A right justified floating point number, the aircraft's maximum rate of descent (feet/minute).
		GS	31-40	A right justified floating point number, the original airspeed of the aircraft.
5	1	IMAX	1-10	A right justified integer, the number of types of weapons, $IMAX \leq 10$.
6	1	WPNAWPNB	1-16	The name of the weapon, e.g., 50 Cal. MG
		JMAX	21-25	A right justified integer, the number of weapons of this type, $JMAX \leq 20$.
		TA	26-30	A right justified integer, the time required to acquire the target by this weapon (seconds), $TA \geq 0$.
		FT	31-35	A right justified integer, the firing time of this weapon before reloading is required (seconds), $FT > 0$.

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
		RLT	36-40	A right justified integer, the time to reload this weapon (seconds) $RLT \geq 0$.
		RMAX	41-50	A right justified floating point number, the maximum range of this weapon(meters) $RMAX > 0$.
		RMIN	51-60	A right justified floating point number, the minimum range of this weapon(meters). $RMIN \geq 0$.
7	2	P	1-8	The probability of kill by this weapon in one time interval at ranges of 100, 200, 300, ..., 1000 meters.
			9-16	
			72-80	
			1-8 9-16 72-80	The second card of this group contains the kill probabilities at ranges of 1100 to 2000 meters in 100 meter intervals.
8	JMAX	NW	1-2	The number assigned to the weapon, $1 \leq NW \leq 99$.
		XW	11-20	The x, y and z coordinates of this weapon, ZW may be blank since PENAIR will compute the weapon's elevation.
		YW	21-30	
		ZW	31-40	

NOTE: Data groups 6, 7 and 8 should be repeated IMAX times.

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
9	1	MMAX	1-10	A right justified integer, the degree of x in the polynomial that approximates the first map square.
		NMAX	11-20	A right justified integer, the degree of y in the polynomial that approximates the first map square
10	As many cards as necessary	PHI	1-80	The coefficients of the first polynomial appear on this and the following cards of this data group in a 4E20.10 field as obtained from the terrain simulation program.

NOTE: Data groups 9 and 10 should be repeated KMAX times.

11	4	AFIELD	1-16	These four cards contain the four lines of the title to be printed at the top of the output.
----	---	--------	------	--

Data Group	Number of Cards	Variable Name	Columns of Cards	Description of Variable
12	1	IMARK	10	<p>The value of this flag, determines the flightpath to be flown.</p> <p>IMARK = 1, upon receiving fire, fly straight and level,</p> <p>= 2, climb straight ahead,</p> <p>= 3, turn away, level,</p> <p>= 4, turn away and climb,</p> <p>= 5, dive and flynap of the earth,</p> <p>= 6, turn away and flynap of the earth,</p> <p>= 7, the flight path is completely pre-determined and read in as data group 13,</p> <p>= 8, the x and y coordinates are read in as data group 13, and a nap of the earth flightpath is computed.</p>
		IPUNCH	20	<p>IPUNCH = 1, the flightpath will be punched on cards.</p> <p>= 0, the flightpath will not be punched.</p>

Data Group	Number of Cards	Variable Name	Column of Card	Description of Variable
		GSNEW	21-30	A right justified floating point number, the ground speed to be attained by the aircraft after receiving fire.

NOTE: Data groups 11 and 12 will be repeated NRUN times.

13	As many cards as necessary but not greater than 300.	NPATH	6-7	Any positive integer, used to identify the flightpath,
		XT	18-26	Right justified floating
		YT	27-35	point numbers, each of the
		ZH	36-44	cards of this group contain x, y, and z coordinates of the aircraft in consecutive time intervals.

NOTE: A blank card should be inserted at the end of this data group. This data group will be used only if IMARK = 7 or 8. If IMARK = 8, ZH may be blank. Data group 13 should follow its corresponding data groups 11 and 12.

Output of the PENAIR Program

A sample output from the PENAIR program is illustrated in Figures 14 and 15. The input parameters of the weapons are printed with the weapon positions and elevations. The second page of the output (Figure 15) is a summary of time in view and aircraft survival probability. The time in view listed here is in number of time intervals in view. If the time interval used in this example had been 0.5 second, the aircraft was in view of weapon number 49, located at (70080, 76940, 339), for 28 seconds (56×0.5). The survival probability of the aircraft from this weapon was .9991.

SUMMARY OF WEAPON DATA

WEAPON TYPE	M14	SQUAD	.50 CAL.	MG	M-42	QUAD	50
NUMBER OF WEAPONS	4	6	4	3	3	3	3
ACQUISITION TIME (SEC)	10	15	6	9000	6	6	6
FIRING TIME (SEC)	6	8	15	100	0	15	15
RELOADING TIME (SEC)	0	100	8	2000	0	100	100
MINIMUM RANGE (METERS)	600	1400	100	2000	0	100	100
MAXIMUM RANGE (METERS)			1400		2000	2000	2000

WEAPON POSITIONS

SITE NUMBER

WEAPON TYPE

M14 SQUAD

1	72810	76150	300
5	71200	76220	315
9	71350	76620	359
12	71020	76450	355

.50 CAL. MG

37	72420	75180	331
38	70240	76050	357
39	71020	76160	357
41	72460	74880	243

M-42

42	71660	74640	311
43	70070	75000	288
49	70020	76940	379

QUAD 50

51	70840	75900	349
52	71620	76000	375

FIGURE 14

FIRST PAGE OF OUTPUT FROM THE PENAIR PROGRAM

MISSION - SUMMARY OF
 RUN NUMBER 2-1-C
 11 APRIL 1965
 FLY STRAIGHT AND LEVEL
 INCREASE SPEED TO 110 KNOTS

SUMMARY OF TIME IN VIEW AND SURVIVAL PROBABILITY BY WEAPON NUMBER

WEAPON NUMBER	1	5	9	12
	37	38	39	41
	42	43	44	
	51	52		

TIME IN VIEW	100	85	38	14
	100	100	100	47
	88	57	56	
	100	100		

SURVIVAL PROBABILITY	1.0000	.6696	.8653	1.0000
	.9611	.9983	.9038	.9869
	.0678	.8168	.9991	
	.7027	.1722		

TOTAL SURVIVAL PROBABILITY .0033

FIGURE 15
 SECOND PAGE OF OUTPUT FROM THE PENAIR PROGRAM

APPENDIX IV

FORTRAN Listing of the TERRAIN Program

TERRAIN consists of a main program and three overlays. The control cards for the COOP-Monitor of the CDC 1604 are included in this listing and are explained in Appendix V. The variable names used in the TERRAIN program are defined in the comments appearing in the FORTRAN listing. The variable names of input and output parameters are explained in Appendix V.


```

-COOP,,HARRISON,O/49/S/56/57/7/E/45=54,45,20000,,PROGRAM TERRAIN.
-BINARY,56.
(MAIN,7.
-FTN,L,E.
      PROGRAM TERRAIN
      COMMON/M1/XFIRST,XLAST,YFIRST,YLAST,Z,IMAX,JMAX
      1/M2/MMAX,NMAX,NEWCTL
      2      /PLOT1/ZTMIN,CND,NC,LABL(50),ITITLE(12)
      3      /PLOT2/ZCOMP(45,45),A,KK
      DIMENSION Z(45,45)
      CALL OVERLAY (7,0,1)
      CALL OVERLAY (7,0,2)

C
      IF(NEWCTL)10,20,10
      NEWCTL=0, NO MAP WILL BE DRAWN.
      =1, A CONTOUR.MAP USING THE APPROXIMATING POLYNOMIAL
      WILL BE DRAWN.

C
      10 CALL OVERLAY(7,0,3)
      20 STOP
      END
      FINIS
-BINARY,56.
(OVERLAY,7,1.
-FTN,L,E.
      PROGRAM MAPPING
      C FINDS THE ELEVATION OF EQUALLY SPACED GRID POINTS.
      C S=SPACING OF GRID LINES A=LINEAR DIMENSION OF MAP RECTANGLE
      C GX=X-COORDINATE OF SW CORNER OF MAP RECTANGLE
      C GY=Y-COORDINATE OF SW CORNER OF MAP RECTANGLE
      C XP(1),YP(1)=MEASURED COORDINATES OF RECTANGLE CORNERS.
      C FZ=HEIGHT OF CONTOUR LINE NC=CARD NO X(I),Y(I)=COORDINATES OF
      C POINTS ON A CONTOUR LINE.
      COMMON/I/S,J,ZC/2/KM/M1/BCX,XLAST,BCY,YLAST,Z,IMAX,JMAX
      1      /PLOT1/ZTMIN,CND,NCI,LABL(50),ITITLE(12)
      2      /PLOT2/ZCOMP(45,45),A ,KK

```

TER0001
TER0002
TER0003
TER0004
TER0005
TER0006
TER0007
TER0008
TER0009
TER0010
TER0011
TER0012

TER0013

TER0014
TER0015
TER0016
TER0017
TER0018
TER0019
TER0020
TER0021
TER0022
TER0023
TER0024
TER0025
TER0026
TER0027
TER0028
TER0029
TER0030
TER0031


```

3      /M2/MMAX,NMAX,NEWCTL
      DIMENSION XC(500),YC(500),YX(45, 70),ZX(45, 70),X(7),Y(7),XP(4),
      1YP(4),XY(45, 70),ZY(45, 70),LX(70),LY(70),YXI(45,70),Z(45,45),
      1XYI(45,70),ZXI(45,70),ZPX(45,45),ZPY(45,45),ZYI(45,70)
      EQUIVALENCE(XC,YXI),(YC,ZXI),(YX,XYI),(ZX,ZYI),(XY,ZPX),(ZY,ZPY)
      1,(LX,Z)
      CORRECT(X,Y,A,X1,X2,X3,Y1) = A*(X-(X-X1))*(Y-Y1)*(X3-X2)/((X3-X1)*
      1(Y3-Y1)-(X1)/(X2-X1)
      DO 300 L=1,50
      300 LABL(L)=4H.
      J=0
      READ 500,ITITLE
      500 FORMAT(6A8)
      READ 700,NL
      700 FORMAT(I5)
      DO 800 L=1,NL
      800 READ 600,II,(LABL(II))
      600 FORMAT(I5,1X,A4)
      READ 900,NMAX,MMAX,NEWCTL
      900 FORMAT(3I5)
      1 READ 100,S,A
      100 FORMAT(2F10)
      READ 200,ZTMIN,ZTMAX,CND
      200 FORMAT(3F10.0)
      NCI=(ZTMAX-ZTMIN)/CND+1.
      KM=(A+200.)/S+1.
      DO 30 K=1,KM $ LX(K)=1
      30 LY(K)=1
      2 READ 101,GX,GY,(XP(I),YP(I),I=1,4)
      101 FORMAT(-3PF2,F2,6X,0P,8F5)
      BCX=GX-100. $ BCY=GY-100.
      4 READ 102,FZ,NC,(X(I),Y(I),I=1,7)
      102 FORMAT(4X,-1PF4,0P,I2,14F5)
      5 IF(NC)14,14,6
      6 GO TO(7,8),NC
      7 ZC=FZ

```

```

TER0032
TER0033
TER0034
TER0035
TER0036
TER0037
TER0038
TER0039

TER0040
TER0041
TER0042
TER0043
TER0044
TER0045
TER0046
TER0047
TER0048
TER0049
TER0050
TER0057
TER0051
TER0052
TER0053
TER0054
TER0055
TER0056
TER0058
TER0059
TER0060
TER0061
TER0062
TER0063
TER0064
TER0065

```



```

      8 DO 13 I=1,7
      10 IF(X(I)+Y(I))12,11,12
      12 XC(I+J)=X(I)
      13 YC(I+J)=Y(I)
        J=J+7
        GO TO 4
      11 J=J+I-1
        GO TO 4
      14 X1=XP(1) $ Y1=YP(1) $ X2=XP(2) $ Y2=YP(2) $ X3=XP(3)
        Y3=YP(3) $ Y4=YP(4)
        DO 17 I=1,J
          XCI=XC(I) $ YCI=YC(I)
      16 XC(I)=CORRECT(XCI,YCI,A,X1,X2,X3,Y1 )+GX
          YC(I)=CORRECT(YCI,XCI,A,Y1,Y4,Y3,X1 )+GY
      17 CONTINUE
      19 CALL CONTOURI(XC,YC,BCX,LX,YX,ZX)
      20 CALL CONTOURI(YC,XC,BCY,LY,XY,ZY)
        IF(FZ.LT.9990.)3,31
      3 READ 102,FZ,NC,(X(I),Y(I),I=1,7)
        J=0
        IF(NC)2,2,7
      31 CALL ORDER(YX,ZX,LX,YXI,ZXI)
      34 CALL ORDER(XY,ZY,LY,XYI,ZYI)
      37 CALL XPOLATE(YXI,ZXI,BCY,LX,ZPX)
      38 CALL XPOLATE(XYI,ZYI,BCX,LY,ZPY)
      39 DO 40 K=1,KM
      40 DO 40 J=1,KM
        Z(K,J)=(ZPX(K,J)+ZPY(J,K))/2.
        IF (Z(K,J).LT.ZTMIN)767,768
      767 Z(K,J)=ZTMIN $ GO TO 40
      768 IF (Z(K,J).GT.ZTMAX)769,40
      769 Z(K,J)=ZTMAX
      40 CONTINUE
        YLAST=BCY+A+200.
        XLAST=BCX+A+200.
        IMAX=KM $ JMAX=KM

```

```

TER0066
TER0067
TER0068
TER0069
TER0070
TER0071
TER0072
TER0073
TER0074
TER0075
TER0076
TER0077
TER0078
TER0079
TER0080
TER0081
TER0082
TER0083
TER0084
TER0085
TER0086
TER0087
TER0088
TER0089
TER0090
TER0091
TER0092
TER0093
TER0094
TER0095
TER0096
TER0097
TER0098
TER0099
TER0100
TER0101

```



```

TER0102
TER0103
TER0104
TER0105
TER0106
TER0107
TER0108
TER0109
TER0110
TER0111
TER0112
TER0113
TER0114
TER0115
TER0116
TER0117
TER0118
TER0119
TER0120
TER0121
TER0122
TER0123
TER0124
TER0125
TER0126
TER0127
TER0128
TER0129
TER0130
TER0131
TER0132
TER0133
TER0134
TER0135
TER0136
TER0137

END
SUBROUTINE CONTOURI(XC,YC,BC,L,YX,ZX)
FINDS THE COORDINATES OF THE TERRAIN AT THE INTERSECTIONS OF
CONTOURS AND GRID LINES.
COMMON/1/S,J,ZC
DIMENSION XC(500),YC(500),L( 70),YX(45, 70),ZX(45, 70)
1 FK=(XC(1)-BC)/S+1. $ KF=FK
2 IF(FK-KF)6,4,6
4 YX(KF,1)=YC(1) $ ZX(KF,1)=ZC $ L(KF)=2
6 DO 11 I=2,J
7 K=(XC(I)-BC)/S+1.
8 IF(K-KF)12,11,10
12 K=K+1 $ KF=K-1 $ GO TO 9
10 KF=K
9 XXK=BC +(K-1)*S
. YX(K,L(K))=(YC(I)-YC(I-1))*(XXK-XC(I-1))/(XC(I)-XC(I-1))+YC(I-1)
ZX(K,L(K))=ZC $ L(K)=L(K)+1
11 CONTINUE
RETURN
END
SUBROUTINE ORDER(YX,ZX,L,YXI,ZXI)
ORDERS THE PAIRS OF VALUES YX(I,J),ZX(I,J) AFTER INCREASING YX(I,J)
C I
COMMON/2/KM
DIMENSION YX(45, 70),ZX(45, 70),L( 70),YXI(45, 70),ZXI(45, 70)
KMI=KM-1
1 DO 10 K=2,KMI $ LK=L(K)-1
2 DO 10 I=1,LK
3 FMIN=YX(K,1) $ MF=1
4 DO 7 M=2,LK
5 IF(YX(K,M)-FMIN)6,6,7
6 FMIN=YX(K,M) $ MF=M
7 CONTINUE
8 YXI(K,1)=FMIN
9 ZXI(K,1)=ZX(K,MF)
10 YX(K,MF)=9999999.

```



```

TER0138
TER0139
TER0140
TER0141
TER0142
TER0143
TER0144
TER0145
TER0146
TER0147
TER0148
TER0149
TER0151
TER0152
TER0153
TER0154
TER0155
TER0156
TER0157
TER0158
TER0159
TER0160
TER0161
TER0162
TER0163
TER0164
TER0165
TER0166
TER0167
TER0168
TER0169
TER0170
TER0171
TER0172
TER0173
TER0174

LK=L(1)-1
DO 20 I=1,LK
  YXI(1,I)=YXI(2,I)
  20 ZXI(1,I)=ZXI(2,I)
  LK=L(KM-1)-1
  DO 30 I=1,LK
    YXI(KM,I)=YXI(KMI,I)
    30 ZXI(KM,I)=ZXI(KMI,I)
  RETURN
END
SUBROUTINE XPOLATE(YXI,ZXI,BCY,LX,ZPX)
C FINDS HEIGHT OF THE TERRAIN AT EQUIDISTANT POINTS ALONG GRID LINES
COMMON/1/S,AA,BB/2/KM
DIMENSION YXI(45,70),ZXI(45,70),LX(70),ZPX(45,45)
1 POL(X,XL,XH,ZL,ZH)=(X-XL)*(ZH-ZL)/(XH-XL)+ZL
10 DO 190 K=1,KM
20 L XK=LX(K)-1 $ KF=K $ M=2
30 IF(LXK-1)60,40,120
40 DO 50 J=1,KM
50 ZPX(K,J)=ZXI(K,1) $ GO TO 190
60 IF(KF-KM/2)80,80,70
70 KF=KF-1 $ GO TO 90
80 KF=KF+1
90 L XK=LX(KF)-1
100 DO 110 J=1,LXK $ ZXI(K,J)=ZXI(KF,J)
110 YXI(K,J)=YXI(KF,J) $ GO TO 30
120 DO 170 J=1,KM
130 YPX=BCY+(J-1)*S
140 DO 160 I=M,LXK
150 IF(YXI(K,I)-YPX)160,170,170
160 CONTINUE
180 I=LXK $ M=I+1
170 ZPX(K,J)=POL(YPX,YXI(K,I-1),YXI(K,I),ZXI(K,I-1),ZXI(K,I))
190 CONTINUE
RETURN
END

```



```

      FINIS
      -BINARY,56.
      (OVERLAY,7,2.
      -FTN,L,E.
      PROGRAM SURFIT
      C   FITS A POLYNOMIAL TO THE POINTS COMPUTED IN OVERLAY 1.
      DIMENSION DENPA(45),DENQA(45),ALPHA(45,45),P(45,45),Q(45,45),
      1PC(45,45),QC(45,45),A(45),B(45),C(45),D(45),X(45),Y(45),
      2Z(45,45),BETA(45,45),PHI(45,45),
      COMMON/M1/XFIRST,XLAST,YFIRST,YLAST,Z,IMAX,JMAX
      2   /PLOT1/ZTMIN,CND,NC,LABL(50),ITITLE(12)
      1   /PLOT2/ZCOMP(45,45),AQ,KK
      1/M2/MMAX,NMAX,NEWCTL
      EQUIVALENCE(P,PC),(Q,QC),(ALPHA,ZCOMP)
      U=1.
      W=1.
      XNUM=IMAX-1
      DELTAX=(XLAST-XFIRST)/XNUM
      II=IMAX-1
      X(1)=XFIRST
      DO 62 I=2,II
      62 X(I)=X(I-1)+DELTAX
      X(IMAX)=XLAST
      YNUM=JMAX-1
      DELTAY=(YLAST-YFIRST)/YNUM
      JJ=JMAX-1
      Y(1)=YFIRST
      DO 72 J=2,JJ
      72 Y(J)=Y(J-1)+DELTAY
      Y(JMAX)=YLAST
      PRINT 2113,ITITLE
      2113 FORMAT(1H1,/////(20X,6A8)////)
      C   PRINT INPUT DATA X,Y,AND Z
      3010 PRINT 2100
      2100 FORMAT(///10X,44HINPUT DATA GRID COORDINATES X DIRECTION X(I)///)
      PRINT2101,( X(I), I=1,IMAX )

```

TER0175
 TER0176
 TER0177
 TER0178
 TER0179

 TER0180
 TER0181
 TER0182
 TER0183
 TER0184
 TER0185
 TER0186
 TER0187
 TER0188
 TER0189
 TER0190
 TER0191
 TER0192
 TER0193
 TER0194
 TER0195
 TER0196
 TER0197
 TER0198
 TER0199
 TER0200
 TER0201
 TER0202
 TER0203
 TER0204
 TER0205
 TER0206
 TER0207
 TER0208
 TER0209


```

2101 FORMAT(4F14.0)
PRINT 2102
2102 FORMAT(/// 10X,44HINPUT DATA GRID COORDINATES Y DIRECTION Y(J)///)
PRINT2101,(Y(J),J=1,JMAX)
PRINT 2103
2103 FORMAT(1H1///20X,33HINPUT DATA GRID ELEVATIONS Z(I,J)///)
PRINT2105,(J, J=1,JMAX)
2105 FORMAT(4H I/J I13,3I14 / (4X,I13,3I14))
DO2106 I=1,IMAX
2106 PRINT2107,I,(Z(I,J),J=1,JMAX)
2107 FORMAT(// 13,4F14.0 /(3X4F14.0))
C NORMALIZATION OF VARIABLES
SUMX=0.
SUMY=0.
SUMZ=0.
DO 10 I=1,IMAX
10 SUMX=SUMX+X(I)
XIMAX=IMAX
XMEANX= SUMX/XIMAX
DO 20 I=1,IMAX
20 X(I)=X(I)-XMEANX
DO 30 J=1,JMAX
30 SUMY=SUMY+Y(J)
XJMAX=JMAX
YMEANY=SUMY/XJMAX
DO 40 J=1,JMAX
40 Y(J)=Y(J)-YMEANY
DO 50 I=1,IMAX
DO 50 J=1,JMAX
50 SUMZ=SUMZ+Z(I,J)
XIJMAX=IMAX*JMAX
ZMEANZ=SUMZ/XIJMAX
DO 60 I=1,IMAX
DO 60 J=1,JMAX
60 Z(I,J)=Z(I,J)-ZMEANZ
C EVALUATE ORTHOGONAL POLYNOMIALS

```

```

TER0210
TER0211
TER0212
TER0213
TER0214
TER0215
TER0216
TER0217
TER0218
TER0219
TER0220
TER0221
TER0222
TER0223
TER0224
TER0225
TER0226
TER0227
TER0228
TER0229
TER0230
TER0231
TER0232
TER0233
TER0234
TER0235
TER0236
TER0237
TER0238
TER0239
TER0240
TER0241
TER0242
TER0243
TER0244
TER0245

```



```

XNUMA=0.
DENA=0.
DO 70 I=1,IMAX
  P(1,I)=1.
  XNUMA=XNUMA+
  X(I)
  70 DENA=DENA+U
  A(2)=XNUMA/DENA
  DO 80 I=1,IMAX
    P(2,I)=X(I)-A(2)
    DO 90 N=3,NMAX
      XNUMA=0.
      DENA=0.
      DENB=0.
      DO 85 I=1,IMAX
        XNUMA=XNUMA+
        X(I)*P(N-1,I)**2
        DENA=DENA+
        P(N-1,I)**2
        85 DENB=DENB+
        P(N-2,I)**2
        A(N)=XNUMA/DENA
        B(N)=DENA/DENB
        DO 90 I=1,IMAX
          P(N,I)=(X(I)-A(N))*P(N-1,I)-B(N)*P(N-2,I)
          XNUMC=0.
          DENC=0.
          DO 100 J=1,JMAX
            Q(1,J)=1.
            XNUMC=XNUMC+
            Y(J)
            100 DENC=DENC+W
            C(2)=XNUMC/DENC
            DO 110 J=1,JMAX
              Q(2,J)=Y(J)-C(2)
              DO 120 M=3,MMAX
                XNUMC=0.
                DENC=0.
                DEND=0.
                DO 115 J=1,JMAX
                  XNUMC=XNUMC+
                  Y(J)*Q(M-1,J)**2

```

```

TER0246
TER0247
TER0248
TER0249
TER0250
TER0251
TER0252
TER0253
TER0254
TER0255
TER0256
TER0257
TER0258
TER0259
TER0260
TER0261
TER0262
TER0263
TER0264
TER0265
TER0266
TER0267
TER0268
TER0269
TER0270
TER0271
TER0272
TER0273
TER0274
TER0275
TER0276
TER0277
TER0278
TER0279
TER0280
TER0281

```



```

115  DENC=DENC+      Q(M-1,J)**2
      DEND=DEND+    Q(M-2,J)**2
      C(M)=XNUMC/DENC
      D(M)=DENC/DEND
      DO 120 J=1,JMAX
120  Q(M,J)=(Y(J)-C(M))*Q(M-1,J)-D(M)*Q(M-2,J)
      DO 130 N=1,NMAX
      DENPA(N)=0.
      DO 130 I=1,IMAX
130  DENPA(N)=DENPA(N)+      P(N,I)**2
      DO 140 M=1,MMAX
      DENQA(M)=0.
      DO 140 J=1,JMAX
140  DENQA(M)=DENQA(M)+      Q(M,J)**2
      ALPH=0.
      DO 150 I=1,IMAX
      DO 150 J=1,JMAX
      ALPH=ALPH+      Z(I,J)*P(N,I)*Q(M,J)
150  CONTINUE
      ALPH(N,M)=ALPH/(DENPA(N)*DENQA(M))
160  BETA(N,M)=ALPHA(N,M)*ALPH
C  APPLICATION OF GAUSS CRITERION TO DETERMINE THE DEGREE POLY WHICH
C  YIELDS THE CLOSEST FIT TO THE GIVEN DATA
      SUMZSQ=0.
      DO 170 I=1,IMAX
      DO 170 J=1,JMAX
170  SUMZSQ=SUMZSQ+      Z(I,J)**2
      IS=1
      IT=1
      DO 190 N=1,NMAX
      BETASUM=0.
      DO 190 M=1,MMAX
      XX=IMAX*JMAX-N*M
      DO 175 L=1,N

```

```

TER0282
TER0283
TER0284
TER0285
TER0286
TER0287
TER0288
TER0289
TER0290
TER0291
TER0292
TER0293
TER0294
TER0295
TER0296
TER0297
TER0298
TER0299
TER0300
TER0301
TER0302
TER0303
TER0304
TER0305
TER0306
TER0307
TER0308
TER0309
TER0310
TER0311
TER0312
TER0313
TER0314
TER0315
TER0316
TER0317

```



```

175 BETASUM=BETASUM+BETA(L,M)
176 IF( BETASUM-SUMZSQ) 172,172,171
177 TRIGAUS=0.
178 GO TO 174
179 IF (IMAX*JMAX-N*M) 2172,171,2172
2172 TRIGAUS=(SUMZSQ-BETASUM)/XX
174 IF(N-1) 176,375,176
375 IF(M-1) 176,177,176
177 GAUSCRT=TRIGAUS
178 GO TO 190
176 IF(GAUSCRT - TRIGAUS) 180,178,180
178 IF(N*M-IS*IT) 179,180,180
179 IS=N
180 IT=M
180 CONTINUE
181 IF( GAUSCRT - TRIGAUS) 190,190,181
181 GAUSCRT=TRIGAUS
182 IS=N
183 IT=M
184 CONTINUE
185 NMAX=IS
186 MMAX=IT
187 TEMP=(IMAX*JMAX-NMAX*MMAX)/(IMAX*JMAX)
188 XMINSD=SQRT( GAUSCRT*TEMP)
189 C EVALUATION OF ORTHO POLY COEFFICIENTS
190 DO 200 N=1,NMAX
191 PC(N,N)=1.
192 NM1=N-1
193 DO 200 IS=1,NM1
194 PC(N,IS) = -A(N)* PC(NM1,IS)
195 IF(IS-1) 197,198,197
197 PC(N,IS)= PC(N,IS) +PC(N-1,IS-1)
198 IF(IS-NM1) 199,200,199
199 PC(N,IS)=PC(N,IS) - B(N)* PC(N-2,IS)
200 CONTINUE
201 DO 210 M=1,MMAX

```

```

TER0318
TER0319
TER0320
TER0321
TER0322
TER0323
TER0324
TER0325
TER0326
TER0327
TER0328
TER0329
TER0330
TER0331
TER0332
TER0333
TER0334
TER0335
TER0336
TER0337
TER0338
TER0339
TER0340
TER0341
TER0342
TER0343
TER0344
TER0345
TER0346
TER0347
TER0348
TER0349
TER0350
TER0351
TER0352
TER0353

```



```

QC(M,M)=1.
MM1=M-1
DO 210 IT=1,MM1
QC(M,IT)= -C(M)* QC(MM1,IT)
IF(IT-1)207,208,207
207 QC(M,IT)= QC(M,IT)+ QC(MM1,IT-1)
208 IF(IT-MM1) 209,210,209
209 QC(M,IT)=QC(M,IT)-D(M)*QC(M-2,IT)
210 CONTINUE
C EVALUATION OF APPROXIMATING POLY COEFF
DO 230 IS=1,NMAX
DO 220 IT=1,MMAX
PHI(IS,IT)=0.
DO 220 N=IS,NMAX
DO 220 M=IT,MMAX
220 PHI(IS,IT)=PHI(IS,IT)+ALPHA(N,M)*PC(N,IS)*QC(M,IT)
230 CONTINUE
C EVALUATE OF DEPENDENT VARIABLES USING THE APPROXIMATING POLY
CMINSQD=0.
CSUMDIF=0.
CMAXDIF=0.
DO 250 I=1,IMAX
DO 250 J=1,JMAX
ZCOMP(I,J)=0.
240 IS=NMAX
248 POLY=PHI(IS,MMAX)
241 IT=MMAX-1
249 POLY=POLY*Y(J)+PHI(IS,IT)
IF(IT-1) 1999,247,242
242 IT=IT-1
GO TO 249
247 ZCOMP(I,J)=ZCOMP(I,J)*X(I)+POLY
243 IF(IS-1) 2000,245,244
244 IS=IS-1
GO TO 248
245 RESCOMP=Z(I,J)-ZCOMP(I,J)

```

```

TER0354
TER0355
TER0356
TER0357
TER0358
TER0359
TER0360
TER0361
TER0362
TER0363
TER0364
TER0365
TER0366
TER0367
TER0368
TER0369
TER0370
TER0371
TER0372
TER0373
TER0374
TER0375
TER0376
TER0377
TER0378
TER0379
TER0380
TER0381
TER0382
TER0383
TER0384
TER0385
TER0386
TER0387
TER0388
TER0389

```



```

Z(I,J)=Z(I,J)+ZMEANZ
ZCOMP(I,J)=ZCOMP(I,J)+ZMEANZ
CMINSQD=CMINSQD+ RESCOMP**2
ABRESCP=ABSF(RESCOMP)
IF(ABRESCP-CMAXDIF) 250,250,246
246 CMAXDIF=ABRESCP
250 CONTINUE
PRINT 1251
1251 FORMAT(1H1///47HCOMPUTED GRID ELEVATIONS Z(I,J) AS EVALUATED BY
1 /12X,24HAPPROXIMATING POLYNOMIAL ///)
PRINT 2105,(J,J=1,JMAX)
DO 260 I=1,IMAX
260 PRINT 2107, I, (ZCOMP(I,J),J=1,JMAX)
CMINSQD=SQRT(CMINSQD/XIJMAX )
CSUMDIF=CSUMDIF/XIJMAX
NMAXM1=NMAX-1
II=I-1
262 PHI(1,1)=PHI(1,1)+ZMEANZ
DO 4001 I=1,IMAX
DO 4001 J=1,JMAX
4001 ZDIFF(I,J)=Z(I,J)-ZCOMP(I,J)
PRINT 4002
4002 FORMAT(1H1///48H TABLE OF DIFFERENCES ACTUAL-COMPUTED ELEVATIONS
1 ///)
PRINT 2105,(J,J=1,JMAX)
DO 4003 I=1,IMAX
4003 PRINT 2107,I,(ZDIFF(I,J),J=1,JMAX)
IF(NEWCTL) 500,501,500
C EVALUATE GRID ELEVATIONS USING THE APPROXIMATING POLYNOMIAL TO BE
C USED IN OVERLAY 3
500 X(1)=XFIRST+100. $ Y(1)=YFIRST+100.
KK=AQ/25+1
DO 502 I=2,KK $ X(I)=X(I-1)+25.
502 Y(I)=Y(I-1)+25.
DO 550 I=1,KK $ DO 550 J=1,KK
ZCOMP(I,J)=0.

```

TER0390
TER0391
TER0392
TER0393
TER0394
TER0395
TER0396
TER0397
TER0398
TER0399
TER0400
TER0401
TER0402
TER0403
TER0404
TER0406
TER0408
TER0410
TER0411
TER0412
TER0413
TER0414
TER0415
TER0416
TER0417
TER0418
TER0419
TER0420
TER0421
TER0422
TER0423
TER0424
TER0425
TER0426
TER0427
TER0428


```

IS=NMAX
548 POLY=PHI( IS,MMAX)
IT=MMAX-1
549 POLY=POLY*(Y(J)-YMEANY)+PHI( IS,IT)
IF(IT-1) 2000,547,542
542 IT=IT-1
GO TO 549
547 ZCOMP(I,J)=ZCOMP(I,J)*(X(I)-XMEANX)+POLY
543 IF( IS-1)2000,550,544
544 IS=IS-1
GO TO 548
550 CONTINUE
C PRINT NEW DEPENDENT VARIABLES
3110 PRINT 551
551 FORMAT(1H1////48HCOMPUTED GRID ELEVATIONS TO BE USED IN OVERLAY 3
1 ///)
PRINT 2105,(J,J=1,KK)
DO 560 I=1,KK
560 PRINT 2107,I,(ZCOMP(I,J),J=1,KK)
501 PRINT 776
776 FORMAT(1H1////40HCOEFFICIENTS OF APPROXIMATING POLYNOMIAL ///)
PRINT 1272
1272 FORMAT( 7H DEGREE 30X 14H DEGREE OF Y// 7H OF X )
1273 FORMAT( 10X,3120)
DO 302 J=1,MMAX
JJ=J-1
PRINT 270,JJ,(PHI(I,J),I=1,NMAX)
270 FORMAT(//15,5X,3E20.10/(10X,3E20.10))
302 CONTINUE
PRINT 1190
1190 FORMAT (///91H MEAN SQUARED COMPUTED MEAN SQUARED
1 MAXIMUM DIFFERENCES
PRINT 1270,XMINSQD,CMINSQD,CMAXDIF
1270 FORMAT (E15.10,5X,E20.10, 7X,E20.10///)
NMAXX=NMAX-1 $ MMAXX=MMAX-1

```

TER0429
TER0430
TER0431
TER0432
TER0433
TER0434
TER0435
TER0436
TER0437
TER0438
TER0439
TER0440
TER0441
TER0442

TER0443
TER0444
TER0445
TER0446
TER0447
TER0448
TER0449
TER0405
TER0450
TER0407
TER0451
TER0452
TER0453
TER0409
TER0454
TER0455
TER0456
TER0457
TER0458
TER0459


```

PUNCH 778, NMAXX, MMAXX
778  FORMAT(2I10)

```

```

PUNCH 777,((PHI(I,J),I = 1,NMAX),J = 1,MMAX)
777  FORMAT(4E20.10)

```

```

GO TO 7777

```

```

2000 PRINT 2001

```

```

2001 FORMAT ( 6H ERROR)

```

```

STOP

```

```

1999 PRINT 1990

```

```

1990 FORMAT(15H IT ERROR STOP )

```

```

STOP

```

```

7777 CONTINUE

```

```

END

```

```

FINIS

```

```

-BINARY,56.

```

```

(OVERLAY,7,3.

```

```

-FTN,L,E.

```

```

PROGRAM PLOTTER

```

```

C PLOTS A MAP USING THE APPROXIMATING POLYNOMIAL.

```

```

COMMON/PLOT1/CNL,CND,NC,LABL(50),ITITLE(12)

```

```

1 /PLOT2/ZGA(45,45),A,KK

```

```

DIMENSION ZG(45,45)

```

```

DO 10 I=1,KK $ DO 10 J=1,KK $ K=KK+1-J

```

```

10 ZG(I,K)=ZGA(I,J)

```

```

CALL CONTOUR(KK,KK,ZG,45,45,CNL,CND,NC,LABL,ITITLE,08.,08.,0)

```

```

END

```

```

SUBROUTINE CONTOUR(MA,NA,AM,MX,NY,CNL,CND,NK,IB,ITITLE,XSC,YSC,

```

```

1KRID)

```

```

DIMENSION AM(MX,NY),IT(12),X(1800),Y(1800),IB(50)

```

```

DIMENSION YMIN(180),XYMAX(180),ITITLE(12)

```

```

COMMON IBODE,LOCMAX,IBLANK,IARGG,IGRID,FLND,FLMD,YSCALE,EXSCALE,

```

```

1 IT,IBEL,MODCURV,ND,MD,XYMAX,YMIN,CL,Y,X

```

```

MD=XMINOF(60,MA)

```

```

MD=XMAXOF(2,MD)

```

```

ND=XMINOF(60,NA)

```

```

ND=XMAXOF(2,ND)

```

TER0460
TER0461
TER0462
TER0463
TER0464
TER0465
TER0466
TER0467
TER0468
TER0469
TER0470
TER0471
TER0472
TER0473
TER0474
TER0475

TER0476
TER0477
TER0478
TER0479
TER0480
TER0483
TER0484
TER0485
TER0486
TER0487
TER0488
TER0489
TER0490
TER0491
TER0492
TER0493
TER0494


```

FLMD=MD
FLND=ND
IF (IB) 51,50,51
50 IQ = -1
GO TO 52
51 IQ = 0
52 IF(NK-50) 18,18,19
19 NC = 50
GO TO 24
18 NC=NK
24 IF(XSC-1.)2,1,1
2 EXSCALE=INTF(FLMD/9.+1.)
3 YSCALE=EXSCALE
GO TO 5
4 YSCALE=INTF(FLND/15.+1.)
EXSCALE=YSCALE
GO TO 5
1 EXSCALE=MIN1F(30.,INTF(XSC))
IF(YSC -1.) 2,16,16
16 YSCALE = MIN1F(30.,INTF(YSC))
IF(FLMD/EXSCALE-9.) 10,10,2
10 IF(FLND/YSCALE-15.) 5,5,2
5 IF(CND) 6,7,6
6 CL=CNL
CNDY = CND
GO TO 12
7 AMAX=-1.E 250
AMIN=1.E250
DO 8 I=1,MD
DO 8 J=1,ND
AA=AM(I,J)
IF(AMAX-AA) 15,9,9
15 AMAX=AA
GO TO 8
9 IF(AMIN-AA) 8,8,11

```

```

TER0495
TER0496
TER0497
TER0498
TER0499
TER0500
TER0501
TER0502
TER0503
TER0504
TER0505
TER0506
TER0507
TER0508
TER0509
TER0510
TER0511
TER0512
TER0513
TER0514
TER0515
TER0516
TER0517
TER0518
TER0519
TER0520
TER0521
TER0522
TER0523
TER0524
TER0525
TER0526
TER0527
TER0528
TER0529
TER0530

```



```

11 AMIN=AA
8  CONTINUE
   CNDY = (AMAX-AMIN) / FLOATF (NC)
   CL=AMIN+.5*CNDY
12  IGRID=KRID
   IBLANK=8H
20  IF(EXSCALE-5.) 20,20,21
21  IF(YSCALE-5.) 22,22,21
21  IQ=-1
22  IBEL = IBLANK
   KBK=0
   DO 40 I = 1,12
   IF (ITITLE(I)) 42,41,42
41  IT(I) = IBLANK
   KBK=KBK+1
   GO TO 40
42  IT(I) = ITITLE(I)
40  CONTINUE
   IF(KBK-12) 60,61,61
61  IT(1)=8HNO NAME
60  XLEN = EXSCALE/24.
   YLEN = YSCALE/24.
   MODCURV=1
   IARGG=0
   DO 13 KLM = 1,NC
   IF(IQ) 45,46,45
46  IBEL=IB(KLM)
45  CALL SCAN (AM,MD,ND,MX,NY,CL)
   CL = CL+CNDY
13  IARGG=1

      C
      FLT1=FLMD+1.
      FLT2 = FLND + 1.
      X(1)=0.
      Y(1)=FLT2
      X(2)=FLT1

```

TER0531
 TER0532
 TER0533
 TER0534
 TER0535
 TER0536
 TER0537
 TER0538
 TER0539
 TER0540
 TER0541
 TER0542
 TER0543
 TER0544
 TER0545
 TER0546
 TER0547
 TER0548
 TER0549
 TER0550
 TER0551
 TER0552
 TER0553
 TER0554
 TER0555
 TER0556
 TER0557
 TER0558
 TER0559
 TER0560
 TER0561
 TER0562
 TER0563
 TER0564
 TER0565
 TER0566


```

Y(2)=FLT2
X(3)=FLT1
Y(3)=0.
IBODE=-1
CALL PLOT (3)
DO 14 KLM=1,LOCMAX
  XXX=XYMAX(KLM) +XLEN/4.
  YYY=YMIN(KLM) +YLEN/4.
  X(1) = XXX
  Y(1) = YYY + YLEN
  X(2) = XXX
  Y(2) = YYY - YLEN
  X(3) = XXX
  Y(3) = YYY
  X(4) = XXX + XLEN
  Y(4) = YYY
  X(5) = XXX - XLEN
  Y(5) = YYY
14 CALL PLOT (5)
MODCURV=3
  X(1)=1.
  Y(1)=FLND
  X(2)=FLMD
  Y(2)=FLND
  X(3)=FLMD
  Y(3)=1.
  IF(IGRID) 32,31,32
32 NX = 3
  GO TO 33
31 X(4)=1.
  Y(4)=1.
  X(5) = 1.
  Y(5) = FLND
  NX = 5
33 CALL PLOT (NX)
  RETURN

```

```

TER0567
TER0568
TER0569
TER0570
TER0571
TER0572
TER0573
TER0574
TER0575
TER0576
TER0577
TER0578
TER0579
TER0580
TER0581
TER0582
TER0583
TER0584
TER0585
TER0586
TER0587
TER0588
TER0589
TER0590
TER0591
TER0592
TER0593
TER0594
TER0595
TER0596
TER0597
TER0598
TER0599
TER0600
TER0601
TER0602

```



```

C
END
SUBROUTINE SCAN (AM, M, N, MX, NY, CL)
C THIS PROGRAM IS WRITTEN BY M.O.DAYHOFF
  DIMENSION AM(MX, NY), REC(900), X(1800), Y(1800)
  DIMENSION IPT(3,3), INX(8), INY(8)
  DIMENSION DUMDUM(384)
  COMMON IBODE, DUMDUM, CV, Y, X, REC, THE, RA, DL, INY, IPT, PY, JT,
1  NQ, NP, IV, IT, ISS, IDY, IDX, IY, IX, NI, NT, MT
  D=0.
  R=1.
  TH=1.570796325
  NP=0
  DL=D
  RA=R
  THE=TH
  MT=M
  NT=N
  CV=CL
  IF (IZW-120631) 1,3,1
1 IPT(1,1)=8
  IPT(1,2)=1
  IPT(1,3)=2
  IPT(2,1)=7
  IPT(2,3)=3
  IPT(3,1)=6
  IPT(3,2)=5
  IPT(3,3)=4
  INX(1)=-1
  INX(2)=-1
  INX(3)=0
  INX(4)=1
  INX(5)=1
  INX(6)=1
  INX(7)=0
  INX(8)=-1

```

```

TER0603
TER0604
TER0605
TER0606
TER0607
TER0608
TER0609
TER0610
TER0611
TER0612
TER0613
TER0614
TER0615
TER0616
TER0617
TER0618
TER0619
TER0620
TER0621
TER0622
TER0623
TER0624
TER0625
TER0626
TER0627
TER0628
TER0629
TER0630
TER0631
TER0632
TER0633
TER0634
TER0635
TER0636
TER0637
TER0638

```


TER0639
 TER0640
 TER0641
 TER0642
 TER0643
 TER0644
 TER0645
 TER0646
 TER0647
 TER0648
 TER0649
 TER0650
 TER0651
 TER0652
 TER0653
 TER0654
 TER0655
 TER0656
 TER0657
 TER0658
 TER0659
 TER0660
 TER0661
 TER0662
 TER0663
 TER0664
 TER0665
 TER0666
 TER0667
 TER0668
 TER0669
 TER0670
 TER0671
 TER0672
 TER0673
 TER0674

```

    INY(1)=0
    INY(2)=1
    INY(3)=+1
    INY(4)=+1
    INY(5)=0
    INY(6)=-1
    INY(7)=-1
    INY(8)=-1
    IZW=120631
      3 XT=MT
        DO 58 J=1,900
          58 REC(J)=0
          ISS=0
          2 MT1=MT-1
            DO 110 I=1,MT1
              IF (AM(I,1)-CV) 55,110,110
              55 IF (AM(I+1,1)-CV)110,57,57
              57 IX=I+1
              IY=1
              IDX=-1
              IDY=0
              CALL TRACE (AM,MX, NY)
              110 CONTINUE
              NT1=NT-1
              DO 20 I=1,NT1
                IF (AM(MT,I)-CV) 15,20,20
                15 IF (AM(MT,I+1)-CV) 20,17,17
                17 IX=MT
                IY=I+1
                IDX=0
                IDY=-1
                CALL TRACE (AM,MX, NY)
                20 CONTINUE
                22 DO 30 I=1,MT1
                  MT2=MT+1-I
                  IF (AM(MT2,NT)-CV) 25,30,30
  
```



```

25 IF (AM(MT2-1,NT)-CV) 30,27,27
27 IX=MT2-1
   IY=NT
   IDX=1
   IDY=0
   CALL TRACE (AM,MX, NY)
30 CONTINUE
   DO 40 I=1,NT1
   NT2=NT+1-I
   IF (AM(1,NT2)-CV) 35,40,40
35 IF (AM(1,NT2-1)-CV) 40,37,37
37 IX=1
   IY=NT2-1
   IDX=0
   IDY=1
   CALL TRACE (AM,MX, NY)
40 CONTINUE
   ISS=1
   NT1=NT-1
   MT1=MT-1
   DO 10 J=2,NT1
   DO 10 I=1,MT1
   IF (AM(I,J)-CV) 5,10,10
5 IF (AM(I+1,J)-CV) 10,7,7
7 COM=100*(I+1)+J
   IF (NP) 12,11,12
12 DO 9 ID=1,NP
9 IF (REC(ID)-COM) 9,10,9
9 CONTINUE
11 IX= I+1
   IY=J
   IDX=-1
   IDY=0
   CALL TRACE (AM,MX, NY)
10 CONTINUE
   RETURN

```

TER0675
 TER0676
 TER0677
 TER0678
 TER0679
 TER0680
 TER0681
 TER0682
 TER0683
 TER0684
 TER0685
 TER0686
 TER0687
 TER0688
 TER0689
 TER0690
 TER0691
 TER0692
 TER0693
 TER0694
 TER0695
 TER0696
 TER0697
 TER0698
 TER0699
 TER0700
 TER0701
 TER0702
 TER0703
 TER0704
 TER0705
 TER0706
 TER0707
 TER0708
 TER0709
 TER0710


```

END
SUBROUTINE TRACE (AM,MY,NY)
  DIMENSION AM(MY,NY),REC(900), X(1800), Y(1800)
  DIMENSION IPT(3,3), INX(8),INY(8)
  DIMENSION DUMDUM(384)
  COMMON IBODE,DUMDUM,CV,Y,X,REC,THE,RA,DL,INY,IPT,PY,JT,
1  N ,NP,IV,IT,ISS,IDY,IDX,IY,IX,NI,NT,MT
  PY=0.0
  RC= COSF(THE)*RA
  RS= SINP(THE)*RA
501 JT=0
  N=0
  IX0=IX
  IY0=IY
  ISX=IDX+2
  ISY=IDY+2
  IS=IPT(ISX,ISY)
  JTB=0
  ISO=IS
  IF(ISO-8)18,18,17
17 ISO=ISO-8
18 IT=0
  5 CONTINUE
  CALL CALC (AM,MY,NY)
  NZ=N
  N=NZ
  IF (IT+JT-1) 49,49,47
47 XS=X(N-1)
  YS=Y(N-1)
  X(N-1)=X(N)
  Y(N-1)=Y(N)
  X(N)=XS
  Y(N)=YS
49 IS=IS+1
  JT=IT
  9 IF (IS-9) 8,7,7

```

```

TER0711
TER0712
TER0713
TER0714
TER0715
TER0716
TER0717
TER0718
TER0719
TER0720
TER0721
TER0722
TER0723
TER0724
TER0725
TER0726
TER0727
TER0728
TER0729
TER0730
TER0731
TER0732
TER0733
TER0734
TER0735
TER0736
TER0737
TER0738
TER0739
TER0740
TER0741
TER0742
TER0743
TER0744
TER0745
TER0746

```



```

7 IS=IS-8
8 IDX=INX(IS)
  IDY=INY(IS)
  IX2=IX+IDX
  IY2=IY+IDY
  JTB=JTB+1
  IF (JTB-1799) 51,51,308
308 PRINT
  103, CV, X(N), Y(N)
103 FORMAT(1H0,23HA CONTOUR LINE AT LEVEL,E12.5,21H WAS TERMINATED AT
  1X=,E12.5,3H Y=,E12.5/
  2 48H BECAUSE IT CONTAINED MORE THAN 1799 PLOT POINTS )
  RETURN
. C
51 IF (JTB-900) 5100,5101,5101
5100 IBODE = 0
  GO TO 5102
5101 IBODE = 1
5102 IF(ISS) 10,10,20
  20 IF(IX-IX0) 12,21,12
  21 IF(IY-IY0) 12,22,12
  22 IF(IS-IS0) 12,23,12
  23 CONTINUE
  CALL CALC (AM,MY,NY)
  GO TO 73
  10 IF(IX2) 13,50,13
  13 IF (IX2-MT) 19,19,50
  19 IF (IY2) 11,50,11
  11 IF (IY2-NT) 12,12,50
  12 IF (CV-AM(IX2,IY2)) 206,206,5
  206 IF (IDX**2+IDY**2-1) 213,6,213
  213 DCP=(AM(IX,IY)+AM(IX2,IY)+AM(IX,IY2)+AM(IX2,IY2))/4.0
  IF (DCP-CV) 5,217,217
  217 IF (INX(IS-1)) 214,215,214
  214 IX=IX+IDX
  IDX=-IDX
  PY=2.0

```

TER0747
 TER0748
 TER0749
 TER0750
 TER0751
 TER0752
 TER0753
 TER0754
 TER0755
 TER0756
 TER0757
 TER0758
 TER0759
 TER0760
 TER0761
 TER0762
 TER0763
 TER0764
 TER0765
 TER0766
 TER0767
 TER0768
 TER0769
 TER0770
 TER0771
 TER0772
 TER0773
 TER0774
 TER0775
 TER0776
 TER0777
 TER0778
 TER0779
 TER0780
 TER0781
 TER0782


```

CALL CALC (AM,MY,NY)
IX=IX+IDX
GO TO 6
215 IY=IY+IDY
IDY=-IDY
PY=2.0
CALL CALC (AM,MY,NY)
IY=IY+IDY
6 IF (AM(IX-1,IY) - CV) 306,16,16
306 NP=NP+1
REC(NP)=100*IX+IY
16 IS=IS+5
IX=IX2
IY=IY2
GO TO 9
50 XT=MT
IF (AM(IX-1,IY)-CV) 307,73,73
307 NP=NP+1
REC(NP)=100*IX+IY
73 DO 74 I=1,N
X(I)=X(I)+RC*Y(I)
74 Y(I)=RS*Y(I)
CALL PLOT ( N)
RETURN
END
SUBROUTINE CALC (AM,MY,NY)
DIMENSION AM(MY,NY),REC(900), X(1800), Y(1800)
DIMENSION IPT(3,3),INX(8),INY(8)
DIMENSION DUMDUM(384)
COMMON IBODE,DUMDUM,CV,Y,X,REC,THE,RA,DL,INY,INX,IPT,PY,JT,
1 N ,NP,IV,IT,ISS,IDY,IDX,IY,IX,NI,NT,MT
IT=0
N=N+1
IF (IDX**2 + IDY**2 -1) 20,1,20
1 IF (IDX) 10,2,10
2 X(N)=IX

```

TER0783
TER0784
TER0785
TER0786
TER0787
TER0788
TER0789
TER0790
TER0791
TER0792
TER0793
TER0794
TER0795
TER0796
TER0797
TER0798
TER0799
TER0800
TER0801
TER0802
TER0803
TER0804
TER0805
TER0806
TER0807
TER0808
TER0809
TER0810
TER0811
TER0812
TER0813
TER0814
TER0815
TER0816
TER0817
TER0818


```

Z=IY
IY2=IY+IDY
DY=IDY
41 Y(N) = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX,IY2))) *DY + Z
RETURN
10 Y(N)=IY
W=IX
DX=IDX
IX2=IX+IDX
44 X(N) = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX2,IY))) *DX +W
RETURN
20 IX2=IX+IDX
IY2=IY+IDY
W=IX
Z=IY
DX=IDX
DY=IDY
DCP=(AM(IX,IY)+AM(IX2,IY)+AM(IX,IY2)+AM(IX2,IY2))/4.0
IF (PY-2.0) 24,21,24
24 IF (DCP-CV) 21,21,25
21 AL=AM(IX,IY)-DCP
23 V=.5*(AL+DCP -CV)/AL
27 X(N)=V*DX+W
Y(N)=V*DY+Z
PY=0.0
RETURN
25 IT=1
AL=AM(IX2,IY2)-DCP
33 V=.5*(AL+DCP-CV)/AL
28 X(N)=-V*DX+W + DX
Y(N)=-V*DY+Z + DY
RETURN
END
SUBROUTINE PLOT (NGG)
DIMENSION X(1800),Y(1800),ITITLE(12),
1 LTITLE(14), KAXIS(5), ICURV(460), JGRID(25), ICONT(1)

```

TER0819
TER0820
TER0821
TER0822
TER0823
TER0824
TER0825
TER0826
TER0827
TER0828
TER0829
TER0830
TER0831
TER0832
TER0833
TER0834
TER0835
TER0836
TER0837
TER0838
TER0839
TER0840
TER0841
TER0842
TER0843
TER0844
TER0845
TER0846
TER0847
TER0848
TER0849
TER0850
TER0851
TER0852
TER0853
TER0854


```

    DIMENSION XMAX(180),YMAX(180),XMIN(180),YMIN(180),XYMAX(180)
    COMMON IBODE,LOCMAX,IBLANK,IARGG,IGRID,FLND,FLMD,YSCALE,EXSCALE,
1  ITITLE,LABEL,MODCURV,ND,MD,XYMAX,YMIN,CL,Y,X,DUMIE(942),ICURV
    DATA (ICONTRL=40000B),(ICURV3=377737720202020B),(ICURV5=
1  1040000000000000B),(IMAS1=1010000000000000B),(IMAS2=
2  1040000000000000B)
C
    NN=NGG
    ICURV4 = IMAS1
    IF (IBODE) 95,4449,67
4449 IF (IARGG) 98,97,4
    97 LOCMAX=0
    4 FINX=X(NN)
    FINY=Y(NN)
    IF(ABSF(FINX-1.)-.00001)61,61,62
C LEFT MARGIN
    61 NN=NN+1
    X(NN) = 1. - .31*EXSCALE
    Y(NN)=FINY
    GOTO 63
    62 IF(ABSF(FINY-1.)-.00001) 69,69,65
C TOP MARGIN
    69 NN=NN+1
    X(NN)=FINX
    Y(NN)=FINY- YSCALE/45.
    GO TO 33
    65 IF(ABSF(FINX-FLMD)-.00001) 70,70,66
C RIGHT MARGIN
    70 NN=NN+1
    X(NN)=FINX+ EXSCALE/45.
    Y(NN)=FINY
    GO TO 63
    66 IF(ABSF(FINY-FLND)-.00001) 68,68,67
C BOTTOM MARGIN
    68 NN=NN+1
    X(NN)=FINX

```

TER0855
 TER0856
 TER0857
 TER0858
 TER0859
 TER0860
 TER0861
 TER0862
 TER0863
 TER0864
 TER0865
 TER0866
 TER0867
 TER0868
 TER0869
 TER0870
 TER0871
 TER0872
 TER0873
 TER0874
 TER0875
 TER0876
 TER0877
 TER0878
 TER0879
 TERC880
 TER0881
 TER0882
 TER0883
 TER0884
 TER0885
 TER0886
 TER0887
 TER0888
 TER0889
 TER0890


```

      Y(NN) = FINY + .31 * YSCALE
C WRITE LABEL VERTICALLY
      GO TO 33
C WRITE LABEL HORIZONTALLY
      63 ICURV4 = IMAS2
      GO TO 33
C INTERNAL CONTOUR
      67 IF(IARGG) 98,95,96
      98 STOP
      96 CYMAX = AMX(NN,Y)
      CYMIN = AMN(NN,Y,IK)
      CXYMAX=X(IK)
      CXMAX = AMX(NN,X)
      CXMIN = AMN (NN,X,IK)
      DO 91 II = 1,LOCMAX
      IK = II
      IF(CYMIN-YMAX(II)) 90,91,91
      90 IF(CYMIN-YMIN(II)) 91,91,92
      92 IF(CXYMAX-XMAX(II)) 93,91,91
      93 IF(CXYMAX-XMIN(II)) 91,91,94
      91 CONTINUE
      LOCMAX=LOCMAX+1
      IK=LOCMAX
      94 XMAX(IK)=CXMAX
      YMAX(IK)=CYMAX
      XMIN(IK)=CXMIN
      YMIN(IK)=CYMIN
      XYMAX(IK)=CXYMAX
      95 LABEL=IBLANK
      GO TO 330
      33 IF(LABEL-IBLANK) 330,331,330
      331 NN = NN-1
      330 IF (IBODE) 81,81,82
      82 NNN = NN-900
      NN = 900
C PROGRAMMER J. R. WARD

```

```

TER0891
TER0892
TER0893
TER0894
TER0895
TER0896
TER0897
TER0898
TER0899
TER0900
TER0901
TER0902
TER0903
TER0904
TER0905
TER0906
TER0907
TER0908
TER0909
TER0910
TER0911
TER0912
TER0913
TER0914
TER0915
TER0916
TER0917
TER0918
TER0919
TER0920
TER0921
TER0922
TER0923
TER0924
TER0925
TER0926

```



```

C      81  NUMPTS = NN
      IF (MODCURV-2) 10,2400,2400
      2400 BACKSPACE 49
      GO TO 240
C      10  ICONT(1) = ICONTRL + 2
      INSERT TITLE SIZE (02B) AHEAD OF MAIN TITLE RECORD.
C      CALL ISHIFT6 (ITITLE, LTITLE)
C      TEST FOR ALL BLANK TITLES.
      DO 9075 I=1,6
      IF (ITITLE(I)-IBLANK) 9074,9075,9074
      9074 IF (ITITLE(I) ) 9080,9075,9080
      9075 CONTINUE
      IT1 = 1
      ICONT(1) = ICONT(1) - 1
      GO TO 9081
      9080 IT1 = 0
      9081 DO 9085 I=7,12
      IF (ITITLE(I) - IBLANK) 9084,9085,9084
      9084 IF (ITITLE(I)) 9090,9085,9090
      9085 CONTINUE
      IT2 = 1
      ICONT(1) = ICONT(1) - 1
      GO TO 9091
      9090 IT2 = 0
C      NOW GENERATE AXES RECORDS.
      9091 LFTMGN = 0
      IBOTMGN = 40
      XFR1 = FLMD / EXSCALE
      JXFR = XFR1
      XFR2 = (FLMD + 1.)/EXSCALE
      JXFR3 = (FLMD-1.)/EXSCALE
      YFR1 = FLND/YSCALE
      JYFR = YFR1
      YFR2 = (FLND+1.)/YSCALE
      IX100 = 100./EXSCALE

```

TER0927
 TER0928
 TER0929
 TER0930
 TER0931
 TER0932
 TER0933
 TER0934
 TER0935
 TER0936
 TER0937
 TER0938
 TER0939
 TER0940
 TER0941
 TER0942
 TER0943
 TER0944
 TER0945
 TER0946
 TER0947
 TER0948
 TER0949
 TER0950
 TER0951
 TER0952
 TER0953
 TER0954
 TER0955
 TER0956
 TER0957
 TER0958
 TER0959
 TER0960
 TER0961
 TER0962


```

IY100 = 100./YSCALE
IXAXIS=0
  IYAXIS = JYFR+1
  ISIZEY = 100. * XFR2
  ISIZEY = 100. * YFR2
  IH = LFTMGN
  JH = IBOTMGN + IYAXIS*100
  LH = ISIZEY
  IHL = 100 * JXFR3 + IX100 - 7
  KAXIS(1) = IPACK12(IH,JH,LH,IHL)
  JHL = JH + 2
  IHL2 = -100
  IVH2 = -EXSCALE
  IVH = -IVH2 * JXFR3 + 1
  KAXIS(2) = IPACK12(JHL,IHL2,IVH,IVH2)
  NH = JXFR3+ 1
  ISH = 8H 14
  IV = LFTMGN
  JV = IBOTMGN + IYAXIS*100 - ISIZEY
  KAXIS(3) = IPACK12(NH,ISH,IV,JV)
  LV = ISIZEY
  IVL = IV - 2
  JVL = JV + LV - 7 -IY100
  JVL2 = -100
  KAXIS(4) = IPACK12(LV,IVL,JVL,JVL2)
  IVV = 1
  INV2 = YSCALE
  INV = (FLND-1.)/YSCALE + 1.
  ISV = 8H 11
  KAXIS(5) = IPACK12(IVV,INV2,INV,ISV)
  NOW GENERATE CURVES.
  SCX = 100./EXSCALE
  SCY = 100./YSCALE
  SHIFTX = IH
  SHIFTY = JH
  240 IF(NUMPTS - 900)241,242,241

```



```

241 SAVE1 = X(NUMPTS + 1)
    SAVE2 = Y(NUMPTS + 1)
    X(NUMPTS + 1) = X(NUMPTS)
    Y(NUMPTS + 1) = Y(NUMPTS)
242 INUM = (NUMPTS + 1)/2
    DO 244 I=1,INUM
        IC1= X(2*I-1)*SCX + SHIFTX
        IC2=-Y(2*I-1)*SCY + SHIFTY
        IC3= X(2*I)*SCX + SHIFTX
        IC4=-Y(2*I)*SCY + SHIFTY
        ICURV(I+1) = IPACK12(IC1,IC2,IC3,IC4)
244 II = I + 3
        IF(NUMPTS - 900)245,246,245
245 X(NUMPTS + 1) = SAVE1
    Y(NUMPTS + 1) = SAVE2
246 CALL IPACKL1(LABEL, LABEL1, IDUMMY)
    ICURV(II-1) = LABEL1
    ICURV(II) = ICURV4
C      NOW WRITE RECORDS.
1260 IF(MODCURV - 1)1260, 1260, 9015.
    CALL IWRITE (ICONT, IDUMMY, ICONT)
    IF(IDUMMY)5000,260,5000
260 IF(II)9269,9268,9269
9268 CALL IWRITE (LTITLE, IDUMMY, LTITLE(7))
    IF(IDUMMY)5000,9269,5000
9269 IF(II2)9271,9270,9271
9270 CALL IWRITE (LTITLE(8), IDUMMY, LTITLE(14))
    IF(IDUMMY)5000,9271,5000
9271 CALL IWRITE (KAXIS, IDUMMY, KAXIS(5))
    IF(IDUMMY)5000,9015,5000
9015 CALL IWRITE (ICURV, IDUMMY, ICURV(II))
    IF(IDUMMY)5000,9020,5000
9020 IF(MODCURV - 1)272,272,9025
272 IF(IGRID - 1)9025,273,9025
C      GENERATE GRID IF CALLED FOR.
273 IBOT = JV + IY100

```

```

TER0999
TER1000
TER1001
TER1002
TER1003
TER1004
TER1005
TER1006
TER1007
TER1008
TER1009
TER1010
TER1011
TER1012
TER1013
TER1014
TER1015
TER1016
TER1017
TER1018
TER1019
TER1020
TER1021
TER1022
TER1023
TER1024
TER1025
TER1026
TER1027
TER1028
TER1029
TER1030
TER1031
TER1032
TER1033
TER1034

```



```

ITOP = JH - IY100
IRHT = LH - IX100
NEXT1 = ITOP
NEXT2 = IRHT
JGRID(1) = 0
DO 1274 J=1,19,2
  JGRID(J+1) = IPACK12 (IX100 , NEXT1, NEXT2, NEXT1)
  IQG = NEXT1- IBOT -100
  IF (IQG) 1275,1275,1290
1290 IF (XABSF(IQG)-1) 1275,1275,1273
1273 NEXT1 = NEXT1 - 100
  JGRID(J+2) = IPACK12 (NEXT2, NEXT1, IX100 , NEXT1)
  IQG = NEXT1- IBOT -100
  IF (IQG) 1276,1276,1291
1291 IF (XABSF(IQG)-1) 1276,1276,1274
1274 NEXT1 = NEXT1 - 100
1275 JGRID(J+2) = IPACK12 (NEXT2, NEXT1, NEXT2, NEXT1)
1276 JGRID(J+3) = ICURV3
  JGRID(J+4) = ICURV5
  CALL IWRITE (JGRID, IDUMMY, JGRID(J+4))
  IF(IDUMMY)5000,1277,5000
1277 NEXT1 = IX100
  NEXT2 = ITOP
  DO 1279 J=1,11,2
    JGRID(J+1) = IPACK12 (NEXT1, IBOT , NEXT1, NEXT2)
    IQG = NEXT1 - IRHT +100
    IF (IQG) 1292,1280,1280
1292 IF (XABSF(IQG)-1) 1280,1280,1278
1278 NEXT1 = NEXT1 + 100
    JGRID(J+2) = IPACK12 (NEXT1, NEXT2, NEXT1, IBOT )
    IQG = NEXT1 - IRHT +100
    IF(IQG) 1293,1281,1281
1293 IF(XABSF(IQG)-1) 1281,1281,1279
1279 NEXT1 = NEXT1 + 100
1280 JGRID(J+2) = IPACK12 (NEXT1, NEXT2, NEXT1, NEXT2)
1281 JGRID(J+3) = ICURV3

```

TER1035
 TER1036
 TER1037
 TER1038
 TER1039
 TER1040
 TER1041
 TER1042
 TER1043
 TER1044
 TER1045
 TER1046
 TER1047
 TER1048
 TER1049
 TER1050
 TER1051
 TER1052
 TER1053
 TER1054
 TER1055
 TER1056
 TER1057
 TER1058
 TER1059
 TER1060
 TER1061
 TER1062
 TER1063
 TER1064
 TER1065
 TER1066
 TER1067
 TER1068
 TER1069
 TER1070


```

JGRID(J+4) = ICURV5
CALL IWRITE (JGRID, IDUMMY, JGRID(J+4))
IF(IDUMMY)5000,9025,5000
C      SET UP RETURN.
9025 IF (MODCURV-3) 87,278,87
278 PRINT 130, (ITITLE(I),I=1,12)
130 FORMAT(21H0CONTOUR GRAPH TITLED/2(5X,6A8)/18H HAS BEEN PLOTTED./
11H0)
IDUMMY = IYP2(IDUMMY)
GO TO 888
C      THESE ARE THE NORMAL RETURNS.
C      NOW SET UP THE RETURN FOLLOWING A TAPE ERROR.
5000 IF(MODCURV - 1)5001,5001,5002
5001 IDUMMY = IYPE1(IDUMMY)
GO TO 1260
5002 PRINT 5100
5100 FORMAT (/, 36H TAPE ERROR IN WRITING GRAPH OUTPUT. )
IDUMMY = IYPE1(IDUMMY)
STOP
C
87 MODCURV=2
888 END FILE 49
IF (EOF,49) 5000,88
88 IF (IBODE) 83,83,84
84 IBODE = 0
NN = NNN
DO 85 I = 1,NN
II = I+900
X(I) = X(II)
85 Y(I) = Y(II)
GO TO 4
C
83 RETURN
END
C      FUNCTION AMX(KK,A)

```

TER1071
TER1072
TER1073
TER1074
TER1075
TER1076
TER1077
TER1078
TER1079
TER1080
TER1081
TER1082
TER1083
TER1084
TER1085
TER1086
TER1087
TER1088
TER1089
TER1090
TER1091
TER1092
TER1093
TER1094
TER1095
TER1096
TER1097
TER1098
TER1099
TER1100
TER1101
TER1102
TER1103
TER1104
TER1105
TER1106


```

DIMENSION A(1800)
F=A(1)
DO 1 II=2,KK
AA=A(II)
IF(F-AA) 2,1,1
2 F=AA
1 CONTINUE
4 AMX=F
RETURN
END
FUNCTION AMN(KK,A,IK)
DIMENSION A(1800)
F=A(1)
IK = 1
DO 1 II=2,KK
AA=A(II)
IF(AA-F) 2,1,1
2 F=AA
IK=II
1 CONTINUE
AMN=F
RETURN
END
C
SUBROUTINE IWRITE(IFIRST, IDUMMY, LAST)
C IFIRST IS THE FIRST VARIABLE TO BE WRITTEN AND ALL VARIABLES FROM
C IT TO LAST WILL BE WRITTEN IN BINARY MODE ON TAPE XX
C IF RECORD CORRECTLY WRITTEN, OTHERWISE SET NON-ZERO. NOTE THAT
I=3
1000 BUFFER OUT(49,1)(IFIRST, LAST)
C TEST FOR WRITE ERRORS AND EOT
2 IF (UNIT,49)2,6,4,5
C EOT OCCURRED
4 IDUMMY=10
GO TO 9
C TAPE WRITE ERROR

```

```

TER1107
TER1108
TER1109
TER1110
TER1111
TER1112
TER1113
TER1114
TER1115
TER1116
TER1117
TER1118
TER1119
TER1120
TER1121
TER1122
TER1123
TER1124
TER1125
TER1126
TER1127
TER1128
TER1129
TER1130
TER1131
TER1132
TER1133
TER1134
TER1135
TER1136
TER1137
TER1138
TER1139
TER1140
TER1141
TER1142

```



```

5 BACKSPACE 49
IF(I) 4,4,66
66 I=I-1
GO TO 1000
C NO ERROR NOEOT, WRITE OKAY
6 IDUMMY=0
9 RETURN
8 IDUMMY=IDUMMY+10
END
C
FUNCTION ITP2 (IDUMMY)
TYPE WORD -CONTOUR-
DATA(IGRAPH=7HCONTOUR)
WRITE(45,1)IGRAPH
ITP2=IDUMMY
1 FORMAT(/ A7)
RETURN
END
C
FUNCTION ITP1 (IDUMMY)
REWIND TAPE 8, REQUEST NEW TAPE, AND WAIT TILL READY.
DIMENSION IOUT( 4 )
DATA ( IOUT=8H PLEASE ,8HLOAD NEW,8H TAPE 8.,8H START. )
END FILE 49
REWIND 49
WRITE(45,1)IOUT
1 FORMAT(/ 4A8)
ITP1=IDUMMY
PAUSE 77777
RETURN
END
C
C
SUBROUTINE ISHIFT6(ITITLE,LTITLE)
INSERTS 02B AHEAD OF 6-WORD TITLE RECORD.
DIMENSION ITITLE(12),LTITLE(14)

```

```

TER1143
TER1144
TER1145
TER1146
TER1147
TER1148
TER1149
TER1150
TER1151
TER1152
TER1153
TER1154
TER1155
TER1156
TER1157
TER1158
TER1159
TER1160
TER1161
TER1162
TER1163
TER1164
TER1165
TER1166
TER1167
TER1168
TER1169
TER1170
TER1171
TER1172
TER1173
TER1174
TER1175
TER1176
TER1177
TER1178

```



```

ICON2=1H2
IBLANK=8H
ENCODE(56,1,LTITLE) ICON2, (ITITLE(I),I=1,6),IBLANK
1 FORMAT (A1,6A8,A7 )
ENCODE(56,1,LTITLE(8)) ICON2,(ITITLE(I),I=7,12),IBLANK
RETURN
END

C
FUNCTION IPACK12(IONE,I2,I3,I4)
PACKS FOUR 12-BIT WORDS INTO ONE 48-BIT WORD.
DATA(ITWOTO35=10000000000000B),(ITWOTO23=1000000000B),(ITWOTO11=1000
10B),(IZER12=7777B)
I11=IONE.AND.IZER12
I22=I2.AND.IZER12
I33=I3.AND.IZER12
I22=I22*ITWOTO23
I33=I33*ITWOTO11
I11=I11*ITWOTO35
IPACK12=I11.OR.I22.OR.I33.OR.I4.AND.IZER12
RETURN
END

C
SUBROUTINE IPACKL1(LABEL1,LABEL2,LABEL3)
UNPACKS AND REPACKS 2 FOUR CHARACTER LABELS
DATA(FLAG1=3777377700000000B)
ENCODE(8,1,LABEL2)LABEL1
1 FORMAT(4X,A4)
LABEL2=FLAG1.OR.LABEL2
ENCODE(8,2,LABEL3)LABEL1
2 FORMAT(4X,R4)
LABEL3=FLAG1.OR.LABEL3
RETURN
END
END
FINIS
-EXECUTE.

```

```

TER1179
TER1180
TER1181
TER1182
TER1183
TER1184
TER1185
TER1186
TER1187
TER1188
TER1189
TER1190
TER1191
TER1192
TER1193
TER1194
TER1195
TER1196
TER1197
TER1198
TER1199
TER1200
TER1201
TER1202
TER1203
TER1204
TER1205
TER1206
TER1207
TER1208
TER1209
TER1210
TER1211
TER1212
TER1213
TER1214

```


APPENDIX V

Inputs and Outputs of the Terrain Simulation Program

The following information is presented for potential users of this program:

A. Tape units

(1) The COOP control cards are shown in the FORTRAN listing (See Appendix IV.) A dash (-) in column one indicates a 7,9 punch, while an open parenthesis () indicates an 11,0,7,9 punch.

(2) Logical tape unit 49 is the graph tape and 45 refers to the type-writer.

(3) Scratch tapes 56 and 57 are necessary and therefore a total of eight tape units are required.

B. Input Formats

The preparation of the input data cards is explained on the following pages of this appendix.

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
1	2	ITITLE	1-48	These two cards contain the two line title desired for the first page of the output and the bottom of the plotted map.
2	1	NL	1-5	A right justified integer, the number of contour lines requiring a label.
3	NL	II	1-5	A right justified integer, the number of the contour level to be labeled, $1 \leq II \leq 50$
		LABL(II)	7-10	A left justified integer, the elevation of contour level II.
4	1	NMAX	1-5	A right justified integer, the maximum degree of x to be considered
		MMAX	6-10	A right justified integer, the maximum degree of y to be considered.
		NEWCTL	11-15	A right justified integer, NEWCTL = 0, no contour map is desired. = 1, an output tape will be generated for the CalComp 165 plotter.

Data Group	Number of Cards	Variable Name	Columns of Cards	Description of Variable
5	1	S	1-10	A right justified floating point number, the spacing desired between grid lines. S must be chosen so that neither the number of vertical nor horizontal grid lines exceed 45.
		A	11-20	A right justified floating point number, the linear dimension of the map square.
6	1	XTMIN	1-10	A right justified floating point number, the elevation of the lowest contour in the map square.
		XTMAX	11-20	A right justified floating point number, the maximum elevation in the map square.
		CND	21-30	A right justified floating point number, the contour interval. The first contour is the value $C = XTMIN$.
7	1	GX	1-2	The first two digits of a five digit floating point number, $GX * 1000 =$ the x coordinate of the lower left corner of the map square.

Data Group	Number of Cards	Variable Name	Columns of Cards	Description of Variation
		GY	3-4	The first two digits of a five digit floating point number , $GY * 1000 =$ the y coordinate of the lower left corner of the map square.
		XP (1)	11-13	A right justified floating point number, the x coordinate of the lower left corner of the map square measured on an arbitrary scale. XP (1) usually should equal 0000 .
		YP(1)	16-20	The y coordinate of the lower left corner measured on an arbitrary scale. YP(1) usually should equal 0000 .
		XP(2)	21-25	The x coordinate of the lower right corner ,
		YP(2)	26-30	The y coordinate of the lower right corner.
		XP (3)	31-35	The x coordinate of the upper right corner.
		XP(3)	36-40	The y coordinate of the upper right corner.
		XP(4)	41-45	The x coordinate of the upper left corner.
		XP(4)	46-50	The y coordinate of the upper left corner.

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
8	As many cards as necessary		1-4	Any identifying code desired Col. 1-4 are not read.
		FZ	5-8	A right justified floating point number, $FZ * 10 =$ the elevation (feet) whose coordinates appear on this card.
		NC	9-10	A sequence number starting at 01 and increasing by one on each card following whose coordinates are those of the same contour line.
		X(1)	11-15	A right justified floating point number, the x coordinate of a point on the contour line.
		Y(1)	16-20	A right justified floating point number, the coordinate of a point on the contour line.
			21-80	F 5.0 fields of additional points on the contour line in the same manner as Col. 11-20.

NOTE: One blank card is inserted after this group of data cards (eighth data group) for each contour line. One additional blank card is inserted if a new seventh data group is read.

Data Group	Number of Cards	Variable Name	Columns of Card	Description of Variable
9	1		7-9	The figure 999 should appear in these columns to signify that this is the last card of the data deck.

NOTE: There should be no blank cards between the last eighth data group and the ninth data group.

Example:

It is desired that grid square 7175 of the U. S. Army Hunter-Liggett Military Reservation be approximated by a polynomial. Figure 17 illustrates the data cards necessary

The words

HLMR CLOSE TERRAIN

GRID SQUARE 7175

will appear at the top of the first page of the printed output and at the bottom of the plotted map. The minimum and maximum elevation appearing in this square are 900 and 1200 feet respectively. The contour interval is 20 feet. The maximum degree of X and Y to be considered is 20. The grid interval will be 50 meters and the square is 1000 meters on a side.

Figure 16 illustrates how the original map was enlarged to read the contour coordinates.

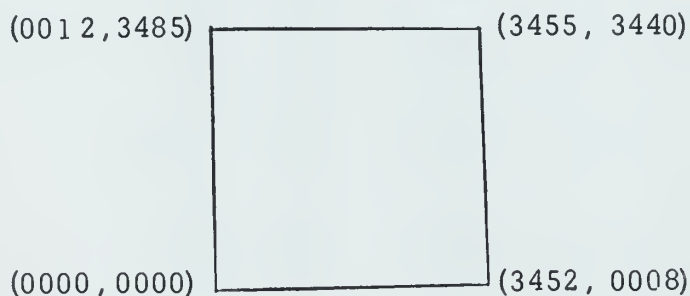


FIGURE 16

DISTORTION CAUSED BY PROJECTION OF A MAP GRID SQUARE

Distortion caused by projection is apparent in Figure 16 and is corrected by the first overlay. The corner coordinates are arbitrary and all contour coordinates are relative to them. If the points are measured by hand, the actual corner coordinates may be used and in this case would

HLMR CLOSE TERRAIN
GRID SQUARE 7175

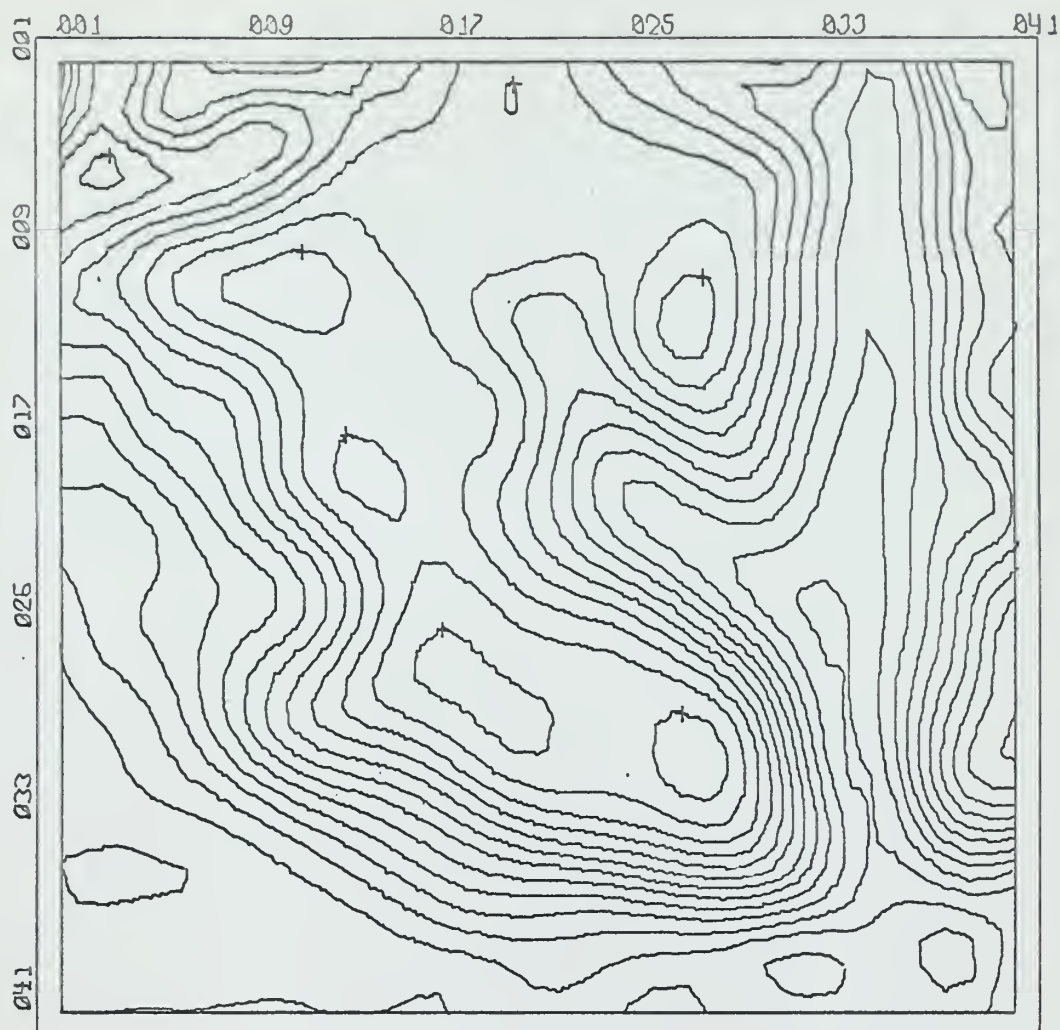
A SAMPLE DATA DECK FOR THE TERRAIN PROGRAM

be (71000, 75000), (72000, 75000), (72000, 76000), (71000, 76000). The contour coordinates would then be the actual terrain coordinates.

In this example the grid square was reprojected a second time with new corner coordinates of (0000, 0000), (3444, -0020), (3441, 3467), (-0004, 3475)

Terrain has three types of output, a) printed, b) punched cards, and c) a plotted map (See Figure 18). The printed output consists of 1) the x and y coordinates of the grid to be used, 2) the input elevations at points on this grid as interpolated by Overlay 1, 3) the elevations of the same points as computed using the approximating polynomial, 4) a table of differences between the actual and computed elevations of the grid, 5) the computed elevations to be used in Overlay 3, 6) the coefficients of the approximating polynomial, and 7) the actual mean elevation squared, computed mean elevation squared and the maximum differences between the actual and computed elevations. Although accuracy is not determined in general, if the actual mean elevation squared and computed mean elevation squared are equal, then computation is exact. In practice they will not be equal due to the imprecise nature of calculation. A wide discrepancy indicates excessive errors in calculation.

The first card of the punched card output contains the degree of x and y in a 2I10 format. The remaining cards contain the coefficients of the approximating polynomial in a 4E20.10 format.



HLMR CLOSE TERRAIN
GRID SQUARE 7174

FIGURE 18

A SAMPLE OF A PLOTTED MAP
FOR THE TERRAIN PROGRAM

thesH2933

A computer simulation of an aircraft pen



3 2768 001 01930 0

DUDLEY KNOX LIBRARY